AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
DESIGN DESCRIPTION

12 June 1992

Prepared By: Intermetrics, Inc.
607 Louis Drive
Warminster, PA 18974

Prepared For: Naval Air Warfare Center - Aircraft Division
Warminster, PA 18974

Contract: N62269-90-C-0412

CDRL: C004

Joseph P. Pettine 6/11/92
Originator **Date**

Stephen C. Nicolay 6/11/92
Reviewer **Date**

AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
DESIGN DESCRIPTION

12 June 1992

Prepared By: Intermetrics, Inc.
607 Louis Drive
Warminster, PA 18974

Prepared For: Naval Air Warfare Center - Aircraft Division
Warminster, PA 18974

Contract: N62269-90-C-0412

CDRL: C004

Joseph P. Pettine 6/11/92
Originator **Date**

Stephen C. Nicolay 6/11/92
Reviewer **Date**

1 Purpose

The Update IV Testbed uses scenario files to control the execution of a test. The scenario file contains Ada code to update the operational database with position and sensor contact data for targets. The testbed executes these calls sequentially as directed causing the appropriate position and contact updates. Normally, the scenario files have to be built manually by placing the correct Ada calls in the file with the data to cause a target position to change, or cause a sensor to report a contact. To cause target movement, each new position must be calculated and entered into the scenario file.

The ATSG automates the scenario file construction process. It will form the correct Ada calls for positioning and sensor contact from the user supplied input data. The input files for ATSG contain movement dynamics and contact information for each target. From this data the ATSG formats the correct Ada call, automatically calculating and updating position based on the movement model given in the input file. Sensor contacts are added as directed by the input file commands.

The ATSG allows multiple scenario files to be generated quickly with little manual calculations. These files can then be used by the Update IV Testbed program which tests Correlation, Collocation, and other functions of the Update IV software.

2 Technical Overview

The main program for the ATSG is atsg_main. This program calls the Input Object to read the input file containing the user's specifications for the Scenario File to be built. It then calls the Event Table Object to merge the Sensor Input, Situation Response, and Operator Action. The result is a time-ordered list of events from the three types of operator input events. Then the Scenario File Object is called from atsg_main to build the Scenario File. Positions are maintained for all targets and the UIV aircraft throughout the exercise by the Platform Movement Model Object.

2.1 Packages

The ATSG is made up of the following packages:

ATSG Generic (ATSG_Gen_Pkg)
Aircraft State Table Object (AC_State_Pkg)
Deterministic Operator Input Table Object (D_OIT_Obj_Pkg)
Deterministic Sensor Input Table Object (D_SIT_Obj_Pkg)
Deterministic Situation Response Table Object (D_SRT_Obj_Pkg)
ESM Table Object (ESM_Table_Obj_Pkg)

Event Table Object (Event_Table_Obj_Pkg)
Free Form Object (Free_Form_Obj_Pkg)
Initialize Scenario File (Init_SF_Pkg)
Input File Object (Input_File_Obj_Pkg)
IRDS Object (IRDS_Object_Out_Pkg)
MAD Object (MAD_Object_Out_Pkg)
Manual Radar Object (Man_Radar_Object_Out_Pkg)
Mark Time Object (Mark_Time_Object_Out_Pkg)
Measurement Data (Measurement_Data_Out_Pkg)
Mark On Top (MOT) (MOT_Object_Out_Pkg)
Object Manager (Object_Manager_Pkg)
Operator (Operator_Pkg)
Pack Strings (Pack_Strings_Pkg)
Platform Movement Model (PLATFORM_MOVEMENT_MODEL)
Plot File (Plot_File_Pkg)
Scenario File Object (Scenario_File_Obj_Pkg)
Schedule (Schedule_Pkg)
Track Data Object (Track_Data_Out_Pkg)
TWS Radar (TWS_Radar_Object_Out_Pkg)
Math Utilities (UT_Math_Pkg)
Visual Object (Visual_Object_Out_Pkg)
Warn (Warn_Pkg)
Write (Write_Files_Pkg)

2.1.1 Package *ATSG Generic* (ATSG_Gen_Pkg)

2.1.1.1 Package Description

This package is used for types used by various packages of the ATSG.

2.1.1.2 Package Data

The following data is defined in the ATSG Generic package. More information for this data can be found in the Data Dictionary.

<u>Type Declarations</u>:
type Event_Count_Type
type Event_Kind_Type

```
type  Model_Kind_Type
type Probability_Type
subtype  Event_Time_Type
subtype Object_ID_Type
subtype Percent_Type
subtype Prob_P_Type is Probability_Type
subtype Prob_Q_Type is Probability_Type
subtype Prob_R_Type is Probability_Type
subtype Target_Index_Type
subtype  Target_Num_Type
```

Data Declarations:

```
Max_Events        :  Event_Count_Type;
Max_Target        : constant := 6;
Max_Time          :  Event_Time_Type;
MAXEVENTS         : constant  := 10000;
Model_Kind        :  Model_Kind_Type;
```

2.1.1.3 Package Units

This package contains no units.

2.1.2 Package *Aircraft State Table Object* (AC_State_Pkg)

2.1.2.1 Package Description

This package writes the AC state table to the scenario file segments.

2.1.2.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

```
Count      : integer;
Quote      : string (1..1);
```

```
Str2      : string (1 .. 20);
Suffix    : Pack_Strings_Pkg.String_Six_Type;
Suffixq   : Pack_Strings_Pkg.String_7_Type;
```

2.1.2.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Init_AC_State
Write_AC_State

2.1.2.3.1 Unit *Init_AC_State*

2.1.2.3.1.1 Purpose

This procedure writes the declaration for TG57_1027_AC_STATE_OBJ to the scenario file.

2.1.2.3.1.2 Input/Output Parameters

None.

2.1.2.3.1.3 Local Data

None.

2.1.2.3.1.4 Processing

Write the task declaration for updating the A/C to the scenario declaration file.

2.1.2.3.1.5 External Interfaces

Called By:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.2.3.1.6 Special Considerations

None.

2.1.2.3.2 Unit *Write_AC_State*

2.1.2.3.2.1 Purpose

This procedure writes the AC state to the scenario file and schedules TG57_1027_AC_STATE_OBJ to update both the AC state and tactical object.

2.1.2.3.2.2 Input/Output Parameters

```
AC_Data      : in PLATFORM_MOVEMENT_MODEL.POSITION_INFO_TYPE;
Event        : in Event_Table_Obj_Pkg.Event_File_Type;
Line_Count   : in out integer
```

2.1.2.3.2.3 Local Data

None.

2.1.2.3.2.4 Processing

Write A/C state record to scenario declaration file.
Write external statement for above declaration.
Add corresponding schedule line to scenario command file.

2.1.2.3.2.5 External Interfaces

Called By:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Pack_Strings_Pkg.Build_Suffix
Pack_Strings_Pkg.Pack_Strings
Pack_Strings_Pkg.Quote_Terminate
Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.2.3.2.6 Special Considerations

None.

2.1.3 Package *Deterministic Operator Input Table Object* (D_OIT_Obj_Pkg)

2.1.3.1 Package Description

This package is used to store the deterministic operator action data for building the Event table.

2.1.3.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type DOIT_Event_Type
type Event_Count_Type
subtype Event_Index_Type
```

Data Declarations:

```
DOIT_Events          : array (Event_Index_Type) of DOIT_Event_Type;
Event_Store_Count    : Event_Count_Type;
Event_Get_Count      : Event_Count_Type:= 1;
Max_DOIT_Events      : constant := 1000;
```

2.1.3.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Get_Next_Event
Store

2.1.3.3.1 Unit *Get_Next_Event*

2.1.3.3.1.1 Purpose

Called by Event_Table_Obj_Pkg.Merge_Tables to determine the next time ordered input event.

2.1.3.3.1.3 Input/Output Parameters

```
Prev_Time     : in  ATSG_Gen_Pkg.Event_Time_Type;
```

```
End_Of_List    : out boolean;
Event          : out DOIT_Event_Type
```

2.1.3.3.1.3 Local Data

```
Searching   : boolean;
```

2.1.3.3.1.4 Processing

Find the next DOIT event to put on the event table and return this event.
The event returned may or may not be put on the event list this call depending on the competition from other event types. If not, the same event would be returned with the next call to this procedure.

2.1.3.3.1.5 External Interfaces

Called by:

Event_Table_Obj_Pkg.Merge_Tables.

Calls:

None.

2.1.3.3.1.6 Special Considerations

None.

2.1.3.3.2 Unit *Store*

2.1.3.3.2.1 Purpose

This unit is called by Input_File_Obj_Pkg.Read_File to store a Deterministic Operator Input Table event in time order.

2.1.3.3.2.3 Input/Output Parameters

```
DOIT_Event     : in DOIT_Event_Type
```

2.1.3.3.2.3 Local Data

None.

2.1.3.3.2.4 Processing

Store a DOIT event on the DOIT list.

2.1.3.3.2.5 External Interfaces

Called by:

Input_File_Obj_Pkg.Read_File.

Calls:

None.

2.1.3.3.2.6 Special Considerations

None.

2.1.4 Package *Deterministic Sensor Input Table Object* (D_SIT_Obj_Pkg)

2.1.4.1 Package Description

This package is used to store the deterministic sensor input for building the Event table.

2.1.4.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

type DSIT_Event_Type
type Event_Count_Type
subtype Event_Index_Type

Data Declarations:

DSIT_Events	: array (Event_Index_Type) of DSIT_Event_Type;
Event_Get_Count	: Event_Count_Type;
Event_Store_Count	: Event_Count_Type;
Max_DSIT_Events	: constant := 1000;

2.1.4.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Get_Next_Event

Store

2.1.4.3.1 Unit *Get_Next_Event*

2.1.4.3.1.1 Purpose

Called by Event_Table_Obj_Pkg.Merge_Tables to determine the next time ordered input event.

2.1.4.3.1.2 Input/Output Parameters

```
Prev_Time     : in ATSG_Gen_Pkg.Event_Time_Type;
End_Of_List   : out boolean;
Event         : out DSIT_Event_Type
```

2.1.4.3.1.3 Local Data

```
Searching  : boolean;
```

2.1.4.3.1.4 Processing

Find the next DSIT event to put on the event table and return this event.

The event returned may or may not be put on the event list this call depending on the competition from other event types. If not, the same event would be returned with the next call to this procedure.

2.1.4.3.1.5 External Interfaces

Called by:

Event_Table_Obj_Pkg.Merge_Tables.

Calls:

None.

2.1.4.3.1.6 Special Considerations

None.

2.1.4.3.2 Unit *Store*

2.1.4.3.2.1 Purpose

This unit is called by Input_File_Obj_Pkg.Read_File to store a Deterministic Sensor Input Table event in time order.

2.1.4.3.2.2 Input/Output Parameters

```
DSIT_Event       : in DSIT_Event_Type
```

2.1.4.3.2.3 Local Data

None.

2.1.4.3.2.4 Processing

Store a DSIT event on the DSIT list.

2.1.4.3.2.5 External Interfaces

Called by:

Input_File_Obj_Pkg.Read_File.

Calls:

None.

2.1.4.3.2.6 Special Considerations

None.

2.1.5 Package *Deterministic Situation Response Table Object* (D_SRT_Obj_Pkg)

2.1.5.1 Package Description

This package is used to store the deterministic situation responses for building the Event table.

2.1.5.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type DSRT_Event_Type
type Event_Count_Type
subtype Event_Index_Type
```

Data Declarations:

```
DSRT_Events          : array (Event_Index_Type) of DSRT_Event_Type;
Event_Get_Count      : Event_Count_Type;
Event_Store_Count    : Event_Count_Type;
Max_DSRT_Events      : constant := 1000;
```

2.1.5.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

```
  Get_Next_Event
  Store
```

2.1.5.3.1 Unit *Get_Next_Event*

2.1.5.3.1.1 Purpose

Called by Event_Table_Obj_Pkg.Merge_Tables to determine the next time ordered situation response event.

2.1.5.3.1.2 Input/Output Parameters

```
  Prev_Time     : in  ATSG_Gen_Pkg.Event_Time_Type;
  End_Of_List   : out boolean;
  Event         : out  DSRT_Event_Type
```

2.1.5.3.1.3 Local Data

```
  Searching   : boolean;
```

2.1.5.3.1.4 Processing

Find the next DSRT event to put on the event table and return this event.

The event returned may or may not be put on the event list this call depending on the competition from other event types. If not, the same event would be returned with the next call to this procedure.

2.1.5.3.1.5 External Interfaces

Called by:

Event_Table_Obj_Pkg.Merge_Tables.

Calls:

None.

2.1.5.3.1.6 Special Considerations

None.

2.1.5.3.2 Unit *Store*

2.1.5.3.2.1 Purpose

This unit is called by Input_File_Obj_Pkg.Read_File to store a Deterministic Situation Response Table event in time order.

2.1.5.3.2.2 Input/Output Parameters

```
DSRT_Event      : in DSRT_Event_Type
```

2.1.5.3.2.3 Local Data

None.

2.1.5.3.2.4 Processing

Store a DSRT event on the DSRT list.

2.1.5.3.2.5 External Interfaces

Called by:

Input_File_Obj_Pkg.Read_File.

Calls:

None.

2.1.5.3.2.6 Special Considerations

None.

2.1.6 Package *ESM Table Object* (ESM_Table_Obj_Pkg)

2.1.6.1 Package Description

This package is used to store the ESM Emitter data for building the Event table and for extracting events from it.

2.1.6.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type Emitter_Count_Type
type Emitter_Event_Type
type Platform_Array_Type
type Target_Table_Type
subtype Emitter_Number_Type
subtype Plat_List_Set_T_type
subtype Platform_Type
subtype Target_Category_Type
subtype Threat_Class_Type
```

Data Declarations:

```
Blank_NATO      : Platform_Type := "         ";
MAXEMITTERS     : constant := 20;
Target_Count    :  ATSG_Gen_Pkg.Target_Num_Type;
Target_Table    : Target_Table_Type;
```

2.1.6.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Get_Data
Store

2.1.6.3.1 Unit *Get_Data*

2.1.6.3.1.1 Purpose

Called by Scenario package to get target and ESM_Data for the given target.

2.1.6.3.1.2 Input/Output Parameters

```
Target_ID   : in   ATSG_Gen_Pkg.TARGET_Index_TYPE;
ESM_Data    : out Emitter_Event_Type
```

2.1.6.3.1.3 Local Data

```
Target_Index     : ATSG_Gen_Pkg.Target_Index_Type;
```

2.1.6.3.1.4 Processing

Select the element corresponding to Target_ID from the Target_Table and output this element as ESM_Data.

2.1.6.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.6.3.1.6 Special Considerations

None.

2.1.6.3.2 Unit *Store*

3.1.6.3.2.1 Purpose

Called by Input_File_Obj_Pkg.Read_File to store an emitter event in the EMITTER_DATA_TABLE.

2.1.6.3.2.2 Input/Output Parameters

```
Emitter_Event   : in Emitter_Event_Type
```

2.1.6.3.2.3 Local Data

None.

2.1.6.3.2.4 Processing

Store an ESM event on the ESM list.

2.1.6.3.2.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.6.3.2.6 Special Considerations

None.

2.1.7 Package *Event Table Object* (Event_Table_Obj_Pkg)

2.1.7.1 Package Description

This package is used to store the event data for building the Event table.

2.1.7.2 Package Data

The following data is defined in the Event Table Object package. More information for this data can be found in the Data Dictionary.

Type Declarations:

type Event_File_Type

type Event_Table_Type

subtype Event_Index_Type

Data Declarations:

2.1.7.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Get_Next_Event

Merge_Tables

2.1.7.3.1 Unit *Get_Next_Event*

2.1.7.3.1.1 Purpose

This procedure is called by Scenario_File.Generate_Scenario_File, to retrieve the next event from the Event Table.

2.1.7.3.1.2 Input/Output Parameters

Event : out Event_File_Type;

2.1.7.3.1.3 Local Data

None.

2.1.7.3.1.4 Processing

Extract the next event from the generalized event list.

2.1.7.3.1.5 External Interfaces

Called by:

Scenario_File.Generate_Scenario_File.

Calls:

None.

2.1.7.3.1.6 Special Considerations

None.

2.1.7.3.2 Unit *Merge_Tables*

2.1.7.3.2.1 Purpose

This procedure is called by the main program to generate a single chronological event list from the individual event lists.

2.1.7.3.2.2 Input/Output Parameters

None.

2.1.7.3.2.3 Local Data

```
type SI_OI_Indicator_Type

Prev_Time          : ATSG_Gen_Pkg.Event_Time_Type;
Time_Now           : ATSG_Gen_Pkg.Event_Time_Type;
SI_OI_Indicator    : SI_OI_Indicator_Type;
DSIT_Event         : D_SIT_Obj_Pkg.DSIT_Event_Type;
DSRT_Event         : D_SRT_Obj_Pkg.DSRT_Event_Type;
DOIT_Event         : D_OIT_Obj_Pkg.DOIT_Event_Type;
ET_Index           : ATSG_Gen_Pkg.Event_Count_Type ;
End_Of_SIT         : boolean;
End_Of_OIT         : boolean;
End_Of_SRT         : boolean;
End_of_Lists       : boolean;
Premature_SRT      : boolean;
```

2.1.7.3.2.4 Processing

Merge the DOIT, DSIT, and DSRT lists together into a single event list.

2.1.7.3.2.5 External Interfaces

Called by:

atsg_main

Calls:

D_OIT_Obj_Pkg.Get_Next_Event
D_SIT_Obj_Pkg.Get_Next_Event
D_SRT_Obj_Pkg.Get_Next_Event

2.1.7.3.2.6 Special Considerations

None.

2.1.8 Package *Free Form Object* (Free_Form_Obj_Pkg)

2.1.8.1 Package Description

This package writes the Free Form object to the scenario file segments.

2.1.8.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

<u>Type Declarations</u>:

type Target_Count_Array_Type
subtype Target_Count_Type

<u>Data Declarations</u>:

Psuffix	: Pack_Strings_Pkg.String_Six_Type;
Quote	: string (1..1) ;
Str2	: string (1 .. 20);
Suffix	: Pack_Strings_Pkg.String_Six_Type;
Suffixq	: Pack_Strings_Pkg.String_7_Type;
Target	: ATSG_Gen_Pkg.Target_Index_Type;
Target_Count_Array	: Target_Count_Array_Type := (others => 0);

2.1.8.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Write_Free_Form_Obj

2.1.8.3.1 Unit *Write_Free_Form_Obj*

2.1.8.3.1.1 <u>Purpose</u>

This unit writes the true target position to the scenario file. The input object is passed to the DBTMAD_MANAGE_ADD_OBJECT_TASK (ID=36) by the scheduling in the scenario file which this procedure writes.

2.1.8.3.1.2 <u>Input/Output Parameters</u>

```
Free_Form_Value     : in
               DBDTBM_FREE_FORM_INCLUDE.DBDTBM_FREE_FORM_REC_TYPE;
Line_Count        : in out integer;
Update_Object_ID : in ATSG_Gen_Pkg.Object_ID_Type
```

2.1.8.3.1.3 Local Data

None.

2.1.8.3.1.4 Processing

Write a free form object in the scenario file based on the Free_Form_Value provided.

2.1.8.3.1.5 External Interfaces

Called by:

atsg_main.

Calls:

Pack_Strings_Pkg.Build_Suffix
Pack_Strings_Pkg.Quote_Terminate
Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.8.3.1.6 Special Considerations

None.

2.1.9 Package *Initialize Scenario File* (Init_SF_Pkg)

2.1.9.1 Package Description

This package supports the scenario file package body by downloading the code to write fixed data to the scenario file including: with/use clauses, types, and tactical initialization data.

2.1.9.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

None.

2.1.9.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Write_CR_Init
Write_General_Types
Write_General_Withs
Write_Tac_Init

2.1.9.3.1 Unit *Write_CR_Init*

2.1.9.3.1.1 Purpose

This routine writes statements to initialize multisensor correlation for the testbed.

2.1.9.3.1.2 Input/Output Parameters

Line_Count : in out integer

2.1.9.3.1.3 Local Data

None.

2.1.9.3.1.4 Processing

Write a declaration for initialization of the multi sensor correlation task in the scenario declaration file.
Write the corresponding schedule task line in the scenario command file.

2.1.9.3.1.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Types

Calls:

Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.9.3.1.6 Special Considerations

None.

2.1.9.3.2 Unit *Write_General_Types*

2.1.9.3.2.1 Purpose

Called by scenario file body to write the type lines always needed.

2.1.9.3.2.2 Input/Output Parameters

Line_Count : in out integer

2.1.9.3.2.3 Local Data

None.

2.1.9.3.2.4 Processing

Write declarations for types generally needed in the scenario declaration file.

2.1.9.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Init_SF_Pkg.Write_CR_Init
Init_SF_Pkg.Write_Tac_Init
Mark_Time_Object_Out_Pkg.Mark_Time_Init
MOT_Object_Out_Pkg.MOT_Init
Object_Manager_Pkg.Build_Operator_Object_Characteristics
Write_Files_Pkg.Write_To_Declare_File

2.1.9.3.2.6 Special Considerations

None.

2.1.9.3.3 Unit *Write_General_Withs*

2.1.9.3.3.1 Purpose

Called by scenario file body to write the with/use lines always needed.

2.1.9.3.3.2 Input/Output Parameters

None.

2.1.9.3.3.3 Local Data

None.

2.1.9.3.3.4 Processing

Write with/use statements generally needed to the scenario file.

2.1.9.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_Scenario_File_WU

2.1.9.3.3.6 Special Considerations

None.

2.1.9.3.4 Unit *Write_Tac_Init*

2.1.9.3.4.1 Purpose

This unit writes the object for initializing the tactical plane in the scenario declaration file.

2.1.9.3.4.2 Input/Output Parameters

```
Line_Count       : in out integer
```

2.1.9.3.4.3 Local Data

```
LAT_GAC                 : Platform_Movement_Model.LAT_TYPE;
LON_GAC                 : Platform_Movement_Model.LON_TYPE;
Quote                   : string (1..1);
Str2 : string (1 .. 20);
TACTICAL_PLANE_REC      : UTGETO_TACTICAL_ORIGIN_INCLUDE.
                                    UTGETO_TACTICAL_ORIGIN_REC_TYPE;
```

2.1.9.3.4.4 Processing

Write the object for initializing the tactical plane in the scenario declaration file. Write the corresponding schedule task line in the scenario command file.

2.1.9.3.4.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Types

Calls:

Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.9.3.4.6 Special Considerations

None.

2.1.10 Package *Input File Object* (Input_File_Obj_Pkg)

2.1.10.1 Package Description

This package processes all ATSG data in the input file.

2.1.10.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Local_Int	: integer
Local_Flt	: float
Target_Number	: ATSG_Gen_Pkg.Target_Num_Type

2.1.10.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Read_DOIT_Event	(separate)
Read_DSIT_Event	(separate)
Read_DSRT_Event	(separate)
Read_Emitter_Event	(separate)
Read_File	
Read_Platform_Data	(separate)
Read_POIT_Event	(separate)
Read_PSIT_Event	(separate)
Read_PSRT_Event	(separate)

2.1.10.3.1 Unit *Read_DOIT_Event*

2.1.10.3.1.1 Purpose

This procedure reads the deterministic operator input data from the input file and produces a DOIT event for DOIT Object if the data is valid.

2.1.10.3.1.2 Input/Output Parameters

Valid_Data : out boolean;
DOIT_Event : out D_OIT_Obj_Pkg.DOIT_Event_Type

2.1.10.3.1.3 Local Data

Action : Operator_Pkg.Action_Type;

2.1.10.3.1.4 Processing

Read and store a DOIT event.

2.1.10.3.1.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.1.6 Special Considerations

None.

2.1.10.3.2 Unit *Read_DSIT_Event*

2.1.10.3.2.1 Purpose

This procedure reads the deterministic sensor input data from the input file and produces a DSIT event for DSIT Object if the data is valid.

2.1.10.3.2.2 Input/Output Parameters

```
Valid_Data    : out boolean;
DSIT_Event    : out D_SIT_Obj_Pkg.DSIT_Event_Type
```

2.1.10.3.2.3 Local Data

None.

2.1.10.3.2.4 Processing

Read and store a DSIT event.

2.1.10.3.2.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.2.6 Special Considerations

None.

2.1.10.3.3 Unit *Read_DSRT_Event*

2.1.10.3.3.1 Purpose

This procedure reads the deterministic situation response data from the input file and produces a DSRT event for DSRT Object if the data is valid.

2.1.10.3.3.2 Input/Output Parameters

```
Valid_Data    : out boolean;
```

DSRT_Event : out D_SRT_Obj_Pkg.DSRT_Event_Type

2.1.10.3.3.3 Local Data

Action : Operator_Pkg.Situation_Response_Type;

2.1.10.3.3.4 Processing

Read and store a DSRT event.

2.1.10.3.3.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.3.6 Special Considerations

None.

2.1.10.3.4 Unit *Read_Emitter_Event*

2.1.10.3.4.1 Purpose

This procedure reads the emitter data from the input file and produces an Emitter_Event for ESM Data Object if the data is valid.

2.1.10.3.4.2 Input/Output Parameters

Valid_Data : out boolean;
Emitter_Event : out ESM_Table_Obj_Pkg.Emitter_Event_Type

2.1.10.3.4.3 Local Data

Threat_Class : ESM_Table_Obj_Pkg.Threat_Class_Type;
Target_Category : ESM_Table_Obj_Pkg.Target_Category_Type;
List_Set_Type : ESM_Table_Obj_Pkg.Plat_List_Set_T_type;
Platform : ESM_Table_Obj_Pkg.Platform_Type;
Dumchar : character;

2.1.10.3.4.4 Processing

Read and store an emitter event.

2.1.10.3.4.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.4.6 Special Considerations

None.

2.1.10.3.5 Unit *Read_File*

2.1.10.3.5.10 Purpose

This procedure reads all ATSG input data from the Input File and distributes it to the appropriate objects.

2.1.10.3.5.2 Input/Output Parameters

None.

2.1.10.3.5.3 Local Data

```
Steering_Command        :  Platform_Movement_Model.INPUT_RECORD_TYPE;
Last_Input_Category     :  Platform_Movement_Model.Input_Category_Type;
Segment_ID              :  Platform_Movement_Model.Segment_ID_Type;
More_Platform_Data      : boolean;
Data_OK                 : boolean;
Steering_Cmd            :  Platform_Movement_Model.INPUT_RECORD_TYPE;
Emitter_Event           :  ESM_Table_Obj_Pkg.Emitter_Event_Type;
PSIT_Event              :  P_SIT_Obj_Pkg.PSIT_Event_Type;
POIT_Event              :  P_OIT_Obj_Pkg.POIT_Event_Type;
PSRT_Event              :  P_SRT_Obj_Pkg.PSRT_Event_Type;
DSIT_Event              :  D_SIT_Obj_Pkg.DSIT_Event_Type;
DOIT_Event              :  D_OIT_Obj_Pkg.DOIT_Event_Type;
DSRT_Event              :  D_SRT_Obj_Pkg.DSRT_Event_Type;
```

```
  Platform_Data            : text_io.file_type;
  Emitter_Data             : text_io.file_type;
  Event_Data               : text_io.file_type;
  ATSG_Input_Value_Error   : exception;
  Last_Time                :  ATSG_Gen_Pkg.Event_Time_Type;
```

2.1.10.3.5.4 Processing

Read and store all DSIT data.
Read and store all DOIT data.
Read and store all DSRT data.
Call MERGE_TABLES in EVENT_TABLE_OBJECT to produce the EVENT_TABLE.

2.1.10.3.5.5 External Interfaces

Called by:
atsg_main

Calls:
D_OIT_Obj_Pkg.Store
D_SIT_Obj_Pkg.Store
D_SRT_Obj_Pkg.Store
ESM_Table_Obj_Pkg.Store
Input_File_Obj_Pkg.Read_DOIT_Event
Input_File_Obj_Pkg.Read_DSIT_Event
Input_File_Obj_Pkg.Read_DSRT_Event
Input_File_Obj_Pkg.Read_Emitter_Event
Input_File_Obj_Pkg.Read_Platform_Event
Input_File_Obj_Pkg.Read_POIT_Event
Input_File_Obj_Pkg.Read_PSIT_Event
Input_File_Obj_Pkg.Read_PSRT_Event
P_OIT_Obj_Pkg.Store
P_SIT_Obj_Pkg.Store
P_SRT_Obj_Pkg.Store
Platform_Movement_Model.STORE_POSITION_AND_STEERING_COMMANDS

2.1.10.3.5.6 Special Considerations

None.

2.1.10.3.6 Unit *Read_Platform_Data*

2.1.10.3.6.1 Purpose

This procedure reads the movement model data from the input file and produces a Steering_Cmd for Platform Movement Model if the data is valid.

2.1.10.3.6.2 Input/Output Parameters

```
Valid_Data        : out boolean;
Steering_Cmd      : out Platform_Movement_Model.INPUT_RECORD_TYPE;
```

2.1.10.3.6.3 Local Data

```
Input_Category          : Platform_Movement_Model.Input_Category_Type;
Lon                     : float;
Altitude_Depth          : float;
Lat                     : float;
Duration                : float;
Heading                 : float;
Speed                   : float range 0.0 .. 99999.0;
AC_DC_Rate              : float;
Radius                  : float;
Direction               : Platform_Movement_Model.Direction_Type;
NO_OF_LOOPS             : integer;
MAN_RDR_SS              : float;
TWS_RDR_SS              : float;
ESM_SS                  : float;
MAD_SS                  : float;
ACOUSTIC_SS             : float;
IRDS_SS                 : float;
VISUAL_SS               : float;
COMM_SS                 : float;
Rad_Per_Deg             : constant := 0.01745329277778;
Input_Category_Error    : exception;
```

2.1.10.3.6.4 Processing

Read and store a platform event for the A/C or target.

2.1.10.3.6.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.6.6 Special Considerations

None.

2.1.10.3.7 Unit *Read_POIT_Event* ***** NOT IMPLEMENTED *****

2.1.10.3.7.1 Purpose

This procedure reads the probabilistic operator input data from the input file and produces a POIT event for POIT Object if the data is valid.

2.1.10.3.7.2 Input/Output Parameters

```
Valid_Data     : out boolean;
POIT_Event     : out P_OIT_Obj_Pkg.POIT_Event_Type
```

2.1.10.3.7.3 Local Data

TBD

2.1.10.3.7.4 Processing

Read and store a POIT event.

2.1.10.3.7.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.7.6 Special Considerations

None.

2.1.10.3.8 Unit *Read_PSIT_Event* ***** NOT IMPLEMENTED *****

2.1.10.3.8.1 Purpose

This procedure reads the probabilistic sensor input data from the input file and produces a PSIT event for PSIT Object if the data is valid.

2.1.10.3.8.2 Input/Output Parameters

```
  Valid_Data        : out boolean;
  PSIT_Event        : out  P_SIT_Obj_Pkg.PSIT_Event_Type
```

2.1.10.3.8.3 Local Data

TBD

2.1.10.3.8.4 Processing

Read and store a PSIT event.

2.1.10.3.8.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.8.6 Special Considerations

None.

2.1.10.3.9 Unit *Read_PSRT_Event* ***** NOT IMPLEMENTED *****

2.1.10.3.9.1 Purpose

This procedure reads the probabilistic situation response data from the input file and produces a PSRT event for PSRT Object if the data is valid.

2.1.10.3.9.2 Input/Output Parameters

```
  Valid_Data        : out boolean;
  PSRT_Event        : out  P_SRT_Obj_Pkg.PSRT_Event_Type
```

2.1.10.3.9.3 Local Data

TBD

2.1.10.3.9.4 Processing

Read and store a PSRT event.

2.1.10.3.9.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

None.

2.1.10.3.9.6 Special Considerations

None.

2.1.11 Package *IRDS Object* (IRDS_Object_Out_Pkg)

2.1.11.1 Package Description

This package writes the IRDS data to the scenario file.

2.1.11.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Quote	: string (1..1)

2.1.11.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

IRDS_Add_Header
IRDS_Out
IRDS_UPD_Header

2.1.11.3.1 Unit *IRDS_Add_Header*

2.1.11.3.1.1 Purpose

This unit prints the IRDS add header to the scenario file.

2.1.11.3.1.2 Input/Output Parameters

None.

2.1.11.3.1.3 Local Data

None.

2.1.11.3.1.4 Processing

Write the header portion of the IRDS object which is specific to the ADD tactical IRDS object.

2.1.11.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.11.3.1.6 Special Considerations

None.

2.1.11.3.2 Unit *IRDS_Out*

2.1.11.3.2.1 Purpose

This routine writes the body portion of the IRDS object to the scenario declaration file.

2.1.11.3.2.2 Input/Output Parameters

IRDS_Value : DBDTEU_TRACK_IRDS_INCLUDE.
DBDTEU_TRACK_IRDS_REC_TYPE

2.1.11.3.2.3 Local Data

Complete_Line : boolean;
Str2 : string (1 .. 20);

2.1.11.3.2.4 Processing

Write the body portion of the IRDS object which is common to both ADD and UPDATE tactical IRDS object.

2.1.11.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Measurement_Data_Out_Pkg.Write_Measurement_Data
Write_Files_Pkg.Write_To_Declare_File

2.1.11.3.2.6 Special Considerations

None.

2.1.11.3.3 Unit *IRDS_UPD_Header*

2.1.11.3.3.1 Purpose

This unit prints the IRDS UPD header to the scenario file.

2.1.11.3.3.2 Input/Output Parameters

ID : ATSG_Gen_Pkg.Object_ID_Type

2.1.11.3.3.3 Local Data

None.

2.1.11.3.3.4 Processing

Write the header portion of the IRDS object which is specific to the UPDATE tactical IRDS object.

2.1.11.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.11.3.3.6 Special Considerations

None.

2.1.12 Package *MAD Object* (MAD_Object_Out_Pkg)

2.1.12.1 Package Description

This package writes the MAD data to the scenario file.

2.1.12.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Quote : string (1..1)

2.1.12.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

MAD_Add_Header
MAD_Out
MAD_UPD_Header

2.1.12.3.1 Unit *MAD_Add_Header*

2.1.12.3.1.1 Purpose

This unit prints the mad add header to the scenario file.

2.1.12.3.1.2 Input/Output Parameters

None.

2.1.12.3.1.3 Local Data

None.

2.1.12.3.1.4 Processing

Write the header portion of the MAD object which is specific to the ADD tactical MAD object.

2.1.12.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.12.3.1.6 Special Considerations

None.

2.1.12.3.2 Unit *MAD_Out*

2.1.12.3.2.1 Purpose

This routine writes the body portion of the MAD object to the scenario declaration file.

2.1.12.3.2.2 Input/Output Parameters

Mad_Value : DBDTEV_TRACK_MAD_INCLUDE.DBDTEV_TRACK_MAD_REC_TYPE

2.1.12.3.2.3 Local Data

Complete_Line	: boolean;
Str2	: string (1 .. 20);

2.1.12.3.2.4 Processing

Write the body portion of the MAD object which is common to both ADD and UPDATE tactical MAD object.

2.1.12.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Measurement_Data_Out_Pkg.Write_Measurement_Data
Write_Files_Pkg.Write_To_Declare_File

2.1.12.3.2.6 Special Considerations

None.

2.1.12.3.3 Unit *MAD_UPD_Header*

2.1.12.3.3.1 Purpose

This unit prints the MAD UPD header to the scenario file.

2.1.12.3.3.2 Input/Output Parameters

ID : ATSG_Gen_Pkg.Object_ID_Type

2.1.12.3.3.3 Local Data

None.

2.1.12.3.3.4 Processing

Write the header portion of the MAD object which is specific to the UPDATE tactical MAD object.

2.1.12.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.12.3.3.6 Special Considerations

None.

2.1.13 Package *Manual Radar Object* (Man_Radar_Object_Out_Pkg)

2.1.13.1 Package Description

This package writes the Manual Radar data to the scenario file.

2.1.13.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Quote : string (1..1)

2.1.13.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Man_Radar_Add_Header
Man_Radar_OUT
Man_Radar_UPD_Header

2.1.13.3.1 Unit *Man_Radar_Add_Header*

2.1.13.3.1.1 Purpose

This unit prints the Manual Radar add header to the scenario file.

2.1.13.3.1.2 Input/Output Parameters

None.

2.1.13.3.1.3 Local Data

None.

2.1.13.3.1.4 Processing

Write the header portion of the Manual Radar object which is specific to the ADD tactical Manual Radar object.

2.1.13.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.13.3.1.6 Special Considerations

None.

2.1.13.3.2 Unit *Man_Radar_Out*

2.1.13.3.2.1 Purpose

This unit writes the body portion of the Manual Radar object to the scenario declaration file.

2.1.13.3.2.2 Input/Output Parameters

Man_Radar_Value :
DBDTEX_TRACK_Manual_Radar_INCLUDE.DBDTEV_TRACK_MAD_REC_TYPE

2.1.13.3.2.3 Local Data

```
Complete_Line  : boolean;
Str2           : string (1 .. 20);
```

2.1.13.3.2.4 Processing

Write the body portion of the Manual Radar object which is common to both ADD and UPDATE tactical Manual Radar object.

2.1.13.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Track_Data_Out_Pkg.Write_Track_Data
Write_Files_Pkg.Write_To_Declare_File

2.1.13.3.2.6 Special Considerations

None.

2.1.13.3.3 Unit *Man_Radar_UPD_Header*

2.1.13.3.3.1 Purpose

This unit prints the Manual Radar UPD header to the scenario file.

2.1.13.3.3.2 Input/Output Parameters

```
ID   : ATSG_Gen_Pkg.Object_ID_Type
```

2.1.13.3.3.3 Local Data

None.

2.1.13.3.3.4 Processing

Write the header portion of the Manual Radar object which is specific to the UPDATE tactical Manual Radar object.

2.1.13.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Track_Data_Out_Pkg.Write_Track_Data
Write_Files_Pkg.Write_To_Declare_File

2.1.13.3.3.6 Special Considerations

None.

2.1.14 Package *Mark Time Object* (Mark_Time_Object_Out_Pkg)

2.1.14.1 Package Description

This package writes the mark time object to the scenario file segments.

2.1.14.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

```
Count           : integer;
Str2            : string (1 .. 20);
Suffix          :  Pack_Strings_Pkg.String_Six_Type;
Suffixq         :  Pack_Strings_Pkg.String_7_Type;
```

2.1.14.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Mark_Time_Init
Mark_Time_OUT

2.1.14.3.1 Unit *Mark_Time_Init*

2.1.14.3.1.1 Purpose

This procedure places initial data on the scenario file for Mark Time

2.1.14.3.1.2 Input/Output Parameters

None.

2.1.14.3.1.3 Local Data

None.

2.1.14.3.1.4 Processing

Write a declaration for the Mark Time task to the scenario declaration file.

2.1.14.3.1.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Types

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.14.3.1.6 Special Considerations

None.

2.1.14.3.2 Unit *Mark_Time_OUT*

2.1.14.3.2.1 Purpose

This procedure prints the mark time object value to the scenario file, and schedules NVNSINPF_MARK_TIME_TASK

2.1.14.3.2.2 Input/Output Parameters

```
Time         : in  ATSG_Gen_Pkg.Event_Time_Type;
Line_Count   : in out integer
```

2.1.14.3.2.3 Local Data

None.

2.1.14.3.2.4 Processing

Write a declaration for Mark Time to the scenario declaration file.

2.1.14.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Pack_Strings_Pkg.Build_Suffix
Pack_Strings_Pkg.Quote_Terminate
Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.14.3.2.6 Special Considerations

None.

2.1.15 Package *Measurement Data* (Measurement_Data_Out_Pkg)

2.1.15.1 Package Description

This package writes the measurement data to the scenario file segments.

2.1.15.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

None.

2.1.15.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Write_Measurement_Data

2.1.15.3.1 Unit *Write_Measurement_Data*

2.1.15.3.1.1 Purpose

This procedure writes the measurement data to the scenario file segments.

2.1.15.3.1.2 Input/Output Parameters

```
MEASUREMENT_DATA      : in CRCRGLCR_CORRELATION_TYPE_INCLUDE.
                                          CRCRGLCR_FIX_RECORD_TYPE
```

2.1.15.3.1.3 Local Data

```
Str2 : string (1 .. 20);
```

2.1.15.3.1.4 Processing

Write the Measurement_Data object to the scenario declaration file.

2.1.15.3.1.5 External Interfaces

Called by:

IRDS_Object_Out_Pkg.IRDS_Out
MAD_Object_Out_Pkg.MAD_OUT

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.15.3.1.6 Special Considerations

None.

2.1.16 Package *Mark On Top (MOT)* (MOT_Object_Out_Pkg)

2.1.16.1 Package Description

This package writes the MOT object to the scenario file.

2.1.16.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

<u>Type Declarations</u>:

None.

<u>Data Declarations</u>:

Count : integer;

2.1.16.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

MOT_Init
MOT_OUT

2.1.16.3.1 Unit *MOT_Init*

2.1.16.3.1.1 <u>Purpose</u>

This procedure places initial data on the scenario file for MOT.

2.1.16.3.1.2 <u>Input/Output Parameters</u>

None.

2.1.16.3.1.3 <u>Local Data</u>

None.

2.1.16.3.1.4 <u>Processing</u>

Write the MOT type to the scenario declaration file.

2.1.16.3.1.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Types

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.16.3.1.6 Special Considerations

None.

2.1.16.3.2 Unit *MOT_OUT*

2.1.16.3.2.1 Purpose

This procedure prints the MOT object value to the scenario file, and schedules task DBADD (DBTMAD_MANAGE_ADD_OBJECT_TASK) = 36.

2.1.16.3.2.2 Input/Output Parameters

```
MOT_Value     :
      DBDTDE_MARK_ON_TOP_INCLUDE.DBDTDE_MARK_ON_TOP_REC_TYPE;
Line_Count    : in out integer
```

2.1.16.3.2.3 Local Data

```
Str20     : string (1 .. 20);
Suffix    : Pack_Strings_Pkg.String_Six_Type;
Suffixq   : Pack_Strings_Pkg.String_7_Type;
```

2.1.16.3.2.4 Processing

Write the MOT object to the scenario declaration file.

2.1.16.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Pack_Strings_Pkg.Build_Suffix
Pack_Strings_Pkg.Quote_Terminate

Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.16.3.2.6 Special Considerations

None.

2.1.17 Package *Object Manager* (Object_Manager_Pkg)

2.1.17.1 Package Description

This package contains types used for the operator input and operator response packages.

2.1.17.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type  Action_Object_ID_Array_Type
type  Response_Object_ID_Array_Type
subtype  Number_Of_Objects_Type
```

Data Declarations:

```
Action_Object_ID_Array        : Action_Object_ID_Array_Type;
Response_Object_ID_Array      : Response_Object_ID_Array_Type;
```

2.1.17.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Build_Operator_Object_Characteristics
Get_Operator_Action_Characteristics
Get_Operator_Response_Characteristics

2.1.17.3.1 Unit *Build_Operator_Object_Characteristics*

2.1.17.3.1.1 Purpose

This procedure builds the arrays for the operator action characteristics.

2.1.17.3.1.2 Input/Output Parameters

None.

2.1.17.3.1.3 Local Data

None.

2.1.17.3.1.4 Processing

Set operator action and response characteristics which cause a tactical object to be created to the number of tactical objects. The default is 0.

2.1.17.3.1.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Types

Calls:

None.

2.1.17.3.1.6 Special Considerations

None.

2.1.17.3.2 Unit *Get_Operator_Action_Characteristics*

2.1.17.3.2.1 Purpose

This function gets the number of additional objects associated with the input action. The return value is of type Number_Of_Objects_Type.

2.1.17.3.2.2 Input/Output Parameters

```
Action     : Operator_Pkg.Action_Type;
```

2.1.17.3.2.3 Local Data

None.

2.1.17.3.2.4 Processing

Extract the number of tactical objects associated with this operator action.

2.1.17.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.17.3.2.6 Special Considerations

None.

2.1.17.3.3 Unit *Get_Operator_Response_Characteristics*

2.1.17.3.3.1 Purpose

This function returns the number of additional objects associated with the response. The return value is of type Number_Of_Objects_Type.

2.1.17.3.3.2 Input/Output Parameters

Response : Operator_Pkg.Situation_Response_Type;

2.1.17.3.3.3 Local Data

None.

2.1.17.3.3.4 Processing

Extract the number of tactical objects associated with this operator response.

2.1.17.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.17.3.3.6 Special Considerations

None.

2.1.18 Package *Operator* (Operator_Pkg)

2.1.18.1 Package Description

This package contains types used for the operator input and operator response packages.

2.1.18.2 Package Data

The following data is defined in the ATSG Generic package. More information for this data can be found in the Data Dictionary.

Type Declarations:

type Action_Type
type Attribute_Type is
type Situation_Response_Type
subtype Comment_Type is string

Data Declarations:

None

2.1.18.3 Package Units

This package contains no units.

2.1.19 Package *Pack Strings* (Pack_Strings_Pkg)

2.1.19.1 Package Description

This package contains string manipulation utilities.

2.1.19.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

subtype String_Six_Type

subtype String_7_Type

<u>Data Declarations</u>:

MAXsize	: constant := 6;
MAXp1	: constant := MAXsize+1;

2.1.19.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Build_Suffix
pack_strings
Quote_Terminate

2.1.19.3.1 Unit *Build_Suffix*

2.1.19.3.1.1 <u>Purpose</u>

This function appends two strings ignoring their first characters separating the two with a '_'. The return value is of type String_Six_Type.

2.1.19.3.1.2 <u>Input/Output Parameters</u>

Left	: string
right	: string

2.1.19.3.1.3 <u>Local Data</u>

Outval	: String_Six_Type;
IOUT	: natural;
ILEFT	: natural;
IRIGHT	: natural;

2.1.19.3.1.4 <u>Processing</u>

Concatenate the two input strings to the output strings ignoring the first character of both and inserting a "_" between them.

Blank fill the output string on the right.

2.1.19.3.1.5 External Interfaces

Called by:

AC_State_Pkg.Write_AC_State
Free_Form_Obj_Pkg.Write_Free_Form_Obj
Mark_Time_Object_Out_Pkg.Mark_Time_OUT
MOT_Object_Out_Pkg.MOT_OUT
Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.19.3.1.6 Special Considerations

None.

2.1.19.3.2 Unit *pack_strings*

2.1.19.3.2.1 Purpose

This function packs two strings into one but omits the first byte of the second string. This is useful for concatenating integers as text to strings since the result of integer'image(integer) has a blank in the first byte. The return value is of type string.

2.1.19.3.2.2 Input/Output Parameters

Left	: string
Right	: string

2.1.19.3.2.3 Local Data

Outval	: string (1 .. Left'last + Right'last-1);
IOUT	: natural;
IRIGHT	: natural;

2.1.19.3.2.4 Processing

Concatenate the two input strings to the output string ignoring the first character of the 2nd string. The output string has a length of 1 less than the sum of the lengths of the two input strings.

2.1.19.3.2.5 External Interfaces

Called by:

AC_State_Pkg.Write_AC_State
Scenario_File_Obj_Pkg.Generate_Scenario_File
Schedule_Pkg.Schedule

Calls:

None.

2.1.19.3.2.6 Special Considerations

None.

2.1.19.3.3 Unit *Quote_Terminate*

2.1.19.3.3.1 Purpose

This function terminates the last non blank character of a string with quotes. The return value is of type String_7_Type.

2.1.19.3.3.2 Input/Output Parameters

Strin	: in String_Six_Type;

2.1.19.3.3.3 Local Data

Not_Terminated	: boolean;
Quote	: character;
Outval	: String_7_Type;

2.1.19.3.3.4 Processing

Search a string right to left for a non blank character.

Copy the input string to the output string appending a double quote after the right most non-blank character.

2.1.19.3.3.5 External Interfaces

Called by:

AC_State_Pkg.Write_AC_State
Free_Form_Obj_Pkg.Write_Free_Form_Obj
Mark_Time_Object_Out_Pkg.Mark_Time_OUT

MOT_Object_Out_Pkg.MOT_OUT
Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:
None.

2.1.19.3.3.6 Special Considerations
None.

2.1.20 Package *Platform Movement Model* (PLATFORM_MOVEMENT_MODEL)

2.1.20.1 Package Description

This package provides for the movement of the P3 UIV and target platform movement and relative position calculations. Operations provided by this package include: Store Position and Steering Commands, Update Latitude/Longitude, Update Slant Range, Update X/Y, Calculate Flat Distance, Update Heading, and Time of Next Steering Command.

2.1.20.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type COMMAND_PTR_TYPE
type COMMAND_TYPE
type DIRECTION_TYPE
type GAC_TYPE
type GAC_XY_TYPE
type INPUT_CATEGORY_TYPE
type INPUT_RECORD_TYPE(INPUT_CATEGORY:
                                        INPUT_CATEGORY_TYPE := LINE)
type PLATFORM_UPDATE_TYPE
type POSITION_INFO_TYPE
type SEG_PTR
type STEERING_CMD_TYPE (COMMAND:COMMAND_TYPE := LINE)
type TARGET_HDG_ARRAY_TYPE
```

type TARGET_POS_ARRAY_TYPE
type UPDATE_TYPE

subtype LAT_TYPE
subtype LON_TYPE
subtype SEGMENT_ID_TYPE
subtype TARGET_NUM_TYPE

<u>**Data Declarations**</u>:

Aircraft	: constant INTEGER;
COMMAND_PTR	: COMMAND_PTR_TYPE
Earth_Radius	: constant FLOAT
END_TIME	: ATSG_Gen_Pkg.Event_Time_Type
GAC	: GAC_TYPE
GAC_XYZ	: GAC_XY_TYPE
INPUT_RECORD	: INPUT_RECORD_TYPE
LAST_ARC_BEGIN_TIME	: FLOAT
LAST_ARC_DIRECTION	: DIRECTION_TYPE
LAST_ARC_DURATION	: FLOAT
LAST_ARC_NO_OF_LOOPS	: INTEGER
LAST_ARC_RADIUS	: FLOAT
LAST_ARC_THETA	: FLOAT
LAST_LINE_HEADING	: FLOAT
LAST_POSITION_EVENT	: PLATFORM_UPDATE_TYPE := (Aircraft .. ATSG_Gen_Pkg.Max_Target => 0.0)
Max_Duration	: constant FLOAT := 864000.0
Max_LAT	: constant FLOAT := PI/2.0 -- radians
Max_LON	: constant FLOAT := PI -- radians
Max_Loop	: constant FLOAT := 100.0
Max_Seg_ID	: constant INTEGER := 50
Max_target	: constant INTEGER := 6
Min_LAT	: constant FLOAT := -PI/2.0 -- radians
Min_LON	: constant FLOAT := -PI -- radians
NEXT_ARC_BEGIN_HEADING	: TARGET_HDG_ARRAY_TYPE
NEXT_SPEED	: FLOAT
PI_over_2	: constant FLOAT := PI / 2.0

PLATFORM_LAST_UPDATE	: PLATFORM_UPDATE_TYPE := (Aircraft .. ATSG_Gen_Pkg.Max_Target => 0.0)
POSITION_INFO	: POSITION_INFO_TYPE
SENSOR_SENSITIVITY	: array(ATSG_gen_Pkg.EVENT_KIND_TYPE) of FLOAT
SS_VALUE_DEFAULT	: array(ATSG_gen_Pkg.EVENT_KIND_TYPE) of BOOLEAN
STEERING_CMD	: array (Aircraft .. ATSG_Gen_Pkg.Max_Target, 1 .. Max_Seg_ID) of STEERING_CMD_TYPE
THIS_BEGIN_TIME	: FLOAT
THIS_DURATION	: FLOAT
THIS_HEADING	: FLOAT
THIS_SPEED	: FLOAT
Two_PI	: constant FLOAT := PI * 2.0

2.1.20.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

COMPUTE_DEFAULT_OFFSET
ESTABLISH_INITIAL_LAT_LON
GET_GAC
MOVE_PLATFORM
STORE_POSITION_AND_STEERING_COMMANDS
UPDATE_ARC
UPDATE_LAT_LON
UPDATE_LINE
UPDATE_PLATFORM_POSITIONS
UPDATE_RANGE_BEARING

2.1.20.3.1 Unit *COMPUTE_DEFAULT_OFFSET*

2.1.20.3.1.1 Purpose

This Module is invoked by MOVE_PLATFORM to compute the perturbed x, y positions when the default sensor sensitivities are to be used. The default sensor sensitivities are taken from BAC software and require unique conversions for each sensor kind.

2.1.20.3.1.2 Input/Output Parameters

```
EVENT_KIND   : in   ATSG_Gen_Pkg.Event_Kind_TYPE;
TARGET       : in   ATSG_Gen_Pkg.Target_Num_Type;
```

2.1.20.3.1.3 Local Data

```
C1                          : constant FLOAT :=  0.008
C2_over_C1                  : constant FLOAT := 0.1 / C1
Constant_A_C                : constant FLOAT := 110.0182  (yards);
Constant_B_C                : constant FLOAT := -0.034965;
Constant_C_C                : constant FLOAT := 0.000025525;
Yards_to_Feet               : constant FLOAT := 3.0;
Y_PRN                       : FLOAT;
X_PRN                       : FLOAT;
SDR                         : FLOAT;
DELTA_X                     : FLOAT;
DELTA_Y                     : FLOAT;
DELTA_Z_SQD                 : FLOAT;
AC_TO_TARGET_SLANT_RNG      : FLOAT;
FLAT_RANGE                  : FLOAT;
DEP_ANGLE                   : FLOAT;
```

2.1.20.3.1.4 Processing

A high-level description of this unit's processing.

2.1.20.3.1.5 External Interfaces

Called by:

PLATFORM_MOVEMENT_MODEL.UPDATE_PLATFORM_POSITIONS

Calls:

random_pkg.g_random

2.1.20.3.1.6 Special Considerations

A target 0 must always exist.

2.1.20.3.2 Unit *ESTABLISH_INITIAL_LAT_LON*

2.1.20.3.2.1 Purpose

This Module is called by INPUT_FILE_OBJECT to map the spherical coordinates (Latitude/Longitude) coordinated into updated GAC. The earth centered reference system is oriented with the X axis pointing from the earth's center through the Greenwich meridian at the equator. The positive Z axis extends through the North Pole. The local horizontal system has a NED (north, east, down) orientation. The altitude in the spherical coordinates are the same as in the planar coordinates.

2.1.20.3.2.2 Input/Output Parameters

TGT_LAT	: in	FLOAT;
TGT_LON	: in	FLOAT;
X_POS	: out	FLOAT;
Y_POS	: out	FLOAT;

2.1.20.3.2.3 Local Data

DIFF_X	: FLOAT;
DIFF_Y	: FLOAT;
DIFF_Z	: FLOAT;
C_INV_1_1	: FLOAT;
C_INV_1_2	: FLOAT;
C_INV_1_3	: FLOAT;
C_INV_2_1	: FLOAT;
C_INV_2_2	: FLOAT;
C_INV_2_3	: FLOAT;

2.1.20.3.2.4 Processing

The spherical coordinates are mapped into the planar x-y coordinates using the relationship described in the header to the code. They are then checked to ensure that they lie within the 300 sq NM tactical plane defined with is origin at GLAT and GLON. If a target is outside the gaming area it is placed at the origin.

2.1.20.3.2.5 External Interfaces

Called by:

PLATFORM_MOVEMENT_MODEL.
STORE_POSITION_AND_STEERING_COMMANDS

Calls:
None.

2.1.20.3.2.6 Special Considerations

None.

2.1.20.3.3 Unit *GET_GAC*

2.1.20.3.3.1 Purpose

This unit is invoked by the SCENARIO FILE to retrieve the latitude and longitude center of the gaming area.

2.1.20.3.3.2 Input/Output Parameters

```
LAT      : out  LAT_TYPE;
LON      : out  LON_TYPE;
```

2.1.20.3.3.3 Local Data

None.

2.1.20.3.3.4 Processing

When called, this routine retrieves the center of the gaming area.

2.1.20.3.3.5 External Interfaces

Called by:
None.

Calls:
None.

2.1.20.3.3.6 Special Considerations

None.

2.1.20.3.4 Unit *MOVE_PLATFORM*

2.1.20.3.4.1 Purpose

This Module is invoked by the UPDATE_PLATFORM_POSITIONS to update the platform (P3 and target) positions data.

2.1.20.3.4.2 Input/Output Parameters

TARGET	: in	ATSG_Gen_Pkg.Target_Num_Type;
EVENT_TIME	: in	ATSG_Gen_Pkg.Event_Time_Type;
EVENT_KIND	: in	ATSG_Gen_Pkg.Event_Kind_TYPE;
GAC_XYZ	: in out	GAC_XY_TYPE;
COMMAND_PTR	: in out	COMMAND_PTR_TYPE

2.1.20.3.4.3 Local Data

NEXT_BEGIN_SEG	: SEGMENT_ID_TYPE;
Y_PRN	: FLOAT;
X_PRN	: FLOAT;

2.1.20.3.4.4 Processing

This routine first finds the pointer to the end of the position update in the steering commands array by sequencing from the start pointer and looking at the end time of each segment until a value of end time is found to be greater than or equal to the update time. If the update time is equal to the end time found, the end pointer points to the segment and the next start pointer is set to the top of the next sequential segment. Once the pointers are all set, MOVE_PLATFORM controls the execution of Update Line and Update Arc depending on the category of each maneuver sequencing from the start pointer segment to the end pointer segment. After the end segment has completed, the start pointer is set equal to the next start pointer determined previously.

2.1.20.3.4.5 External Interfaces

Called by:

PLATFORM_MOVEMENT_MODEL.UPDATE_PLATFORM_POSITIONS

Calls:

COMPUTE_DEFAULT_OFFSET
UPDATE_LINE
UPDATE_ARC

g_random

2.1.20.3.4.6 Special Considerations

A target 0 must always exist.

2.1.20.3.5 Unit *STORE_POSITION_AND_STEERING_COMMANDS*

2.1.20.3.5.1 Purpose

This Module is invoked by the INPUT_FILE_OBJECT to store the Initial positional data and steering commands for the P3 UIV aircraft and each of the targets.

2.1.20.3.5.2 Input/Output Parameters

```
INPUT_RECORD       : in  INPUT_RECORD_TYPE;
```

2.1.20.3.5.3 Local Data

```
TARGET_LAT          : array (Aircraft .. ATSG_Gen_Pkg.Max_Target) of LAT_TYPE;
TARGET_LON          : array (Aircraft .. ATSG_Gen_Pkg.Max_Target) of LON_TYPE;
Degrees_to_Radians  : constant FLOAT := PI / 180.0;
Knots_to_ft_per_sec : constant FLOAT := 6076.0 / 3600.0;
NM_to_ft            : constant FLOAT := 6076.0 / 1.0;
Two_PI              : constant FLOAT := PI * 2.0;
First               : constant INTEGER := 1;
DELTA_ANGLE         : FLOAT;
THIS_AC_DC_RATE     : FLOAT;
```

2.1.20.3.5.4 Processing

If the segment it is called for an initialization category it then calls ESTABLISH_INITIAL_LAT_LON routine to place the targets in the local coordinate system.

If the Category is LINE it stores the update position command for the platform specified. The End of Maneuver time is computed and stored with the update position time as:

$$\text{begin\_time} = \text{last\_begin\_time} + \text{last\_duration} \qquad \{1\}$$

where:

begin_time is the time from the beginning of the scenario to the beginning of this segment. The beginning time of the of the first segment is set to zero. (sec)
last_begin_time is the begin_time of the last segment. (sec)
last_duration is the duration of the last segment. (sec)

For a line, the duration is simply taken from the input record for the line.

For an arc, the duration is computed using:

duration = 1 / speed * (4PI * radius * num_loops + (delta_heading) * radius) {2}

where:

duration is the duration of this segment. (sec)
speed is the speed for this arc segment. (ft/sec)
4PI * radius is the circumference of a circle. (ft)
delta_heading is the difference between heading of for the end_heading for this arc segment and the heading of the last line segment. (radians)
radius is the radius of the turn (ft)

For a line segment, the speed is computed from the speed in the input record as follows:

speed [ft/sec] = speed [NM/hour] * 1 [hour] / 3600 [sec] * 6076 [ft] / 1 [NM]
= speed [NM/hour] * 1.68777778 [ft*hour/sec*NM] {3}

The line segment heading is converted from degrees to radians using:

heading [rad] = heading [deg] * PI/180 [rad/deg] {4}

The Arc end heading is defined by the following line's heading. The radius of the arc is converted from NM to feet by:

radius [ft] = radius [NM] * 6076 [ft/NM] {5}

The Arc speed is defined by the previous line's speed, duration and acceleration.

arc_speed [ft/sec] = speed [ft/sec] + accel [ft/sec**2] * duration [sec] {6}

Theta is set for the arc commands as an interpretation of the direction of the turn. Theta for a right turn is PI/2(radians) while it is -PI/2 for a left turn. Also, for the arc segment the segment executed flag is set to false.

2.1.20.3.5.5 External Interfaces

Called by:

Input_File_Obj_Pkg.READ_FILE

Calls:

ESTABLISH_INITIAL_LAT_LONG

2.1.20.3.5.6 Special Considerations

A target 0 must always exist.

2.1.20.3.6 Unit *UPDATE_ARC*

2.1.20.3.6.1 Purpose

This Module is called by MOVE_PLATFORM to update GAC x-y coordinates of each platform based on an arc maneuver. Speed is the speed to be used for this update (ft/sec).

2.1.20.3.6.2 Input/Output Parameters

```
EVENT_TIME        : in     ATSG_Gen_Pkg.Event_Time_Type;
SPEED             : in     FLOAT;
RADIUS            : in     FLOAT;
HEADING           : in out FLOAT;
THETA             : in     FLOAT;
X_POS             : in out FLOAT;
Y_POS             : in out FLOAT;
PLATFORM          : in     ATSG_Gen_Pkg.Target_Num_Type;
```

2.1.20.3.6.3 Local Data

```
X_ORG                 : FLOAT;
Y_ORG                 : FLOAT;
RADIUS_TO_TURN_ORG    : FLOAT;
BETA                  : FLOAT;
ALFA                  : FLOAT;
DELTA_TIME            :  ATSG_Gen_Pkg.Event_Time_Type;
```

2.1.20.3.6.4 Processing

The delta time for this update is calculated using:

delta_time = event_time - begin_time {1}

where:

delta_time is the length of the update (sec)

begin_time is the time of the last update (sec)

event_time is the current time (sec)

The coordinates of the turn origin are computed:

beta = heading - theta {2}

x_org = x_pos - r * cos(beta) {3}

y_org = x_pos - r * sin(beta) {4}

alfa = (theta / abs(theta)) * delta_time * speed / radius {5}

heading = begin_heading + alfa

{6}

where:

x_org is the x coordinate of the turn origin

y_org is the y coordinate of the turn origin

alfa and beta are intermediate calculations

heading is the heading of the platform entering the arc

theta is -PI/2 or PI/2 set depending of the direction of the turn (left or right).

x is the x position of the platform

y is the y position of the platform

r is the radius of the arc (ft)

Then the x,y position of the platform at the end of this arc update is calculated by:

x_pos = x_org + r * cos(beta + alfa) {7}

y_pos = y_org + r * sin(beta + alfa) {8}

After the x and y positions are updated, the beginning segment time is set not to the event time. The arc updates are always computed from the beginning of the arc segment to eliminate the need to calculate intermediate begin headings.

2.1.20.3.6.5 <u>External Interfaces</u>

Called by:

MOVE_PLATFORM

Calls:

None.

2.1.20.3.6.6 Special Considerations

A target 0 must always exist.

2.1.20.3.7 Unit *UPDATE_LAT_LON*

2.1.20.3.7.1 Purpose

This Module is called by UPDATE_PLATFORM_POSITIONS to map the updated GAC coordinates into spherical coordinates (Latitude/Longitude). The mapping method from the horizontal plane to the spherical earth is gnomonic projection. The earth centered reference system is oriented with the X axis pointing from the earth's center through the Greenwich meridian at the equator. The positive Z axis extends through the North Pole. The local horizontal system has a NED (north, east, down) orientation. The altitude in the spherical coordinates are the same as in the planar coordinates.

2.1.20.3.7.2 Input/Output Parameters

X_POS	: in	FLOAT;
Y_POS	: in	FLOAT;
LAT	: out	LAT_TYPE;
LON	: out	LON_TYPE;

2.1.20.3.7.3 Local Data

XPOS	: FLOAT;
YPOS	: FLOAT;
E_LAT	: FLOAT;
E_LON	: FLOAT;
C1	: FLOAT;
C2	: FLOAT;
C3	: FLOAT;
X_Y_PROJ	: FLOAT;

2.1.20.3.7.4 Processing

The following relationships are used to map the planar x-y coordinates into the spherical coordinates:

$$C1 = -\mathrm{Cos}(GLON) * \mathrm{Sin}(GLAT) * xpos - \mathrm{Sin}(GLON) * ypos + R * \mathrm{Cos}(GLAT) * \mathrm{Cos}\ (GLON) \quad \{1\}$$

$$C2 = -\mathrm{Sin}(GLON) * \mathrm{Sin}(GLAT) * xpos + \mathrm{Cos}(GLON) * ypos + R * \mathrm{Cos}(GLAT) * \mathrm{Sin}\ (GLON) \quad \{2\}$$

$$C3 = \mathrm{Cos}(GLAT) * xpos + R * \mathrm{Sin}(GLAT) \quad \{3\}$$

Where:

C1 = An intermediate result of the rotation.
C2 = An intermediate result of the rotation.
C3 = An intermediate result of the rotation.
R = The Earths radius = 20925640.0 ft
GLAT = Gaming Area Center Latitude (radians)
GLON = Gaming Area Center Longitude (radians)
xpos = Platform X-position.
ypos = Platform Y-position.

Using C1 and C2, the longitude of the platform is computed using:

$$LON = \mathrm{Arc\ Tan}(C1/C2) \quad \{4\}$$

Then the Latitude is calculated based on the value of the longitude.

- If the Sin(LON) is equal to 0 then the:

  $$x\_y\_proj = C1 \quad \{5\}$$

- If the Sin(LON) is not equal to 0 then the:

  $$x\_y\_proj = C2\ /\ \mathrm{Sin}(LON) \quad \{6\}$$

Once the projection is determined the Latitude is calculated using:

$$LAT = \mathrm{Arc\ Tan}\ (C3\ /\ x\_y\_proj) \quad \{7\}$$

Where:

LAT = Platform Latitude (radians)
LON = Platform Longitude (radians)
C1 = An intermediate result of the rotation.
C2 = An intermediate result of the rotation.
x_y_proj = The projection into the x-y plane.
C3 = An intermediate result of the rotation.

2.1.20.3.7.5 External Interfaces

Called by:

PLATFORM_MOVEMENT_MODEL.UPDATE_PLATFORM_POSITIONS

Calls:

utmama_math_utils_pkg.UTMAMA_ARCTAN2_FUNC
utmama_math_utils_pkg.UTMAMA_FLOAT

2.1.20.3.7.6 Special Considerations

None.

2.1.20.3.8 Unit *UPDATE_LINE*

2.1.20.3.8.1 Purpose

This Module is called by MOVE_PLATFORM to update the GAC x-y coordinates of each platform.

2.1.20.3.8.2 Input/Output Parameters

EVENT_TIME	: in	ATSG_Gen_Pkg.Event_Time_Type;
AC_DC_RATE	: in	FLOAT;
HEADING	: in	FLOAT;
SPEED	: in out	FLOAT;
X_POS	: in out	FLOAT;
Y_POS	: in out	FLOAT;
PLATFORM	: in	ATSG_Gen_Pkg.Target_Num_Type;

2.1.20.3.8.3 Local Data

MANEUVER_EX_TIME	: ATSG_Gen_Pkg.Event_Time_Type;
END_SPEED	: FLOAT;
AVE_SPEED	: FLOAT;
Half_speed	: constant FLOAT := 0.5;

2.1.20.3.8.4 Processing

First the Speed of the platform is updated using:

$$maneuver\_execution\_time = event\_time - begin\_time \quad \{1\}$$

old_speed = speed {2}
speed = old_speed + old_speed * accel_deccel_rate * maneuver_execution_time {3}
ave_speed = 0.5 * (old_speed + speed) {4}

where:

speed is the speed to of the platform at the end of the maneuver. (ft/sec)
ave_speed is the speed to be used for this update (ft/sec)
old_speed is the speed to be used for the last update (ft/sec)
accel_deccel_rate is the acceleration/deceleration rate specified for this maneuver. (ft/sec**2)
maneuver_execution_time is the length of time for this maneuver. (sec)

The target X and Y Positions are then updated using:

x_pos = old_x_pos + cos(heading) * ave_speed {5}
y_pos = old_y_pos + sin(heading) * ave_speed {6}

where:

x_pos is the updated x coordinate (ft)
y_pos is the updated y coordinate (ft)
old_x_pos is the last updated x coordinate (ft)
old_y_pos is the last updated y coordinate (ft)
heading is the platform north reference heading (radians)
ave_speed is the speed to be used for this update (ft/sec)

After the x and y positions are update the beginning segment time is set to the event time.

2.1.20.3.8.5 External Interfaces

Called by:

MOVE_PLATFORM

Calls:

None.

2.1.20.3.8.6 Special Considerations

A target 0 must always exist.

2.1.20.3.9 Unit *UPDATE_PLATFORM_POSITIONS*

2.1.20.3.9.1 Purpose

This Module is invoked by the SCENARIO_FILE_OBJECT to update the platform (P3 and target) positional data each event. This process controls the updates of the A/C and target platforms.

2.1.20.3.9.2 Input/Output Parameters

```
TARGET          : in    ATSG_Gen_Pkg.Target_Num_Type;
EVENT_TIME      : in    ATSG_Gen_Pkg.Event_Time_Type;
EVENT_KIND      : in    ATSG_Gen_Pkg.Event_Kind_Type;
R_GEO_ERROR     : in    BOOLEAN;
POS_INFO        : out   POSITION_INFO_TYPE;
```

2.1.20.3.9.3 Local Data

```
Two_NM     :constant FLOAT := 6076.0 * 2.0;
Y_PRN      : FLOAT;
X_PRN      : FLOAT;
```

2.1.20.3.9.4 Processing

If the A/C needs to be updated, it calls the MOVE_PLATFORM module for platform 0. The Aircraft will need to be updated for each new event time. When the platform positions are updated, then UPDATE_RANGE_BEARING is invoked to calculate the relative positions between the P3 and the targets. Finally, the gaming area coordinates are mapped into spherical earth coordinates by the UPDATE_LAT_LON procedure.

2.1.20.3.9.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

PLATFORM_MOVEMENT_MODEL.COMPUTE_DEFAULT_OFFSET
PLATFORM_MOVEMENT_MODEL.MOVE_PLATFORM
PLATFORM_MOVEMENT_MODEL.UPDATE_LAT_LON
PLATFORM_MOVEMENT_MODEL.UPDATE_RANGE_BEARING
random_pkg.g_random;

2.1.20.3.9.6 Special Considerations

A target 0 must always exist.

2.1.20.3.10 Unit *UPDATE_RANGE_BEARING*

2.1.20.3.10.1 Purpose

This Module is called by UPDATE_PLATFORM_POSITIONS after the GAC coordinates are updated to calculate the flat distance between P3 UIV aircraft and the target.

2.1.20.3.10.2 Input/Output Parameters

Z_POS_Aircraft	: in	FLOAT;
Z_POS_PLATFORM	: in	FLOAT;
X_POS_Aircraft	: in	FLOAT;
X_POS_PLATFORM	: in	FLOAT;
Y_POS_Aircraft	: in	FLOAT;
Y_POS_PLATFORM	: in	FLOAT;
AC_TO_TARGET_SLANT_RNG	: out	FLOAT;
AC_TO_TARGET_BEARING	: out	FLOAT;
AC_TO_TARGET_FLAT_RNG	: out	FLOAT;

2.1.20.3.10.3 Local Data

DELTA_X	: FLOAT;
DELTA_Y	: FLOAT;
ADJ	: FLOAT;
DELTA_Z_SQD	: FLOAT;
FLAT_RANGE	: FLOAT;

2.1.20.3.10.4 Processing

The flat range is calculated using:

```
delta_x = p3_x_pos - target_x_pos
delta_y = p3_y_pos - target_y_pos
flat_range = (delta_x**2 + delta_y**2)))0.5
```

where:

flat_range is the horizontal distance between the A/C and the target. (feet)

delta_x is the signed distance from aircraft to target.
delta_x is the signed distance from aircraft to target.
p3_x_pos is the x coordinate of the P3. (feet)
target_x_pos is the x coordinate of the target. (feet)
p3_y_pos is the y coordinate of the P3. (feet)
target_y_pos is the y coordinate of the target. (feet)

As the altitude and depth of the platforms are constant the delta in altitude can be initialized as:

delta_z_sqd = (p3_z_pos - target_z_pos)**2

where:

delta_z_sqd is the difference in the z positions squared.
p3_z_pos is the p3 altitude
target_z_pos is the target altitude/depth.

Using the flat range and delta z squared:

slant_range = [(flat_range)**2 + delta_z_sqd]**0.5

where:

slant_range is the actual distance from the A/C to the target. (feet)
flat_range is the horizontal distance between the A/C and the target. (feet)
delta_z_sqd is the difference in the z positions squared.

The bearing form the aircraft to the target with respect to true north is then calculated. The equation for the bearing is dependent on the relative positions of the target and the aircraft. First an intermediate angle adj is calculated using:

adj = arc cos (abs (delta_y)) / flat_range

The adj angle is then used in all four cases to determine the bearing.

- If the Aircraft is north east of the target then given: delta_x > 0 and delta_y > 0 :

  Bearing = PI - adj

- If the Aircraft is north west of the target then given: delta_x > 0 and delta_y < 0 :

  Bearing = PI + adj

- If the Aircraft is south east of the target then given: delta_x < 0 and delta_y > 0 :
  Bearing = adj

- If the Aircraft is south west of the target then given: delta_x < 0 and delta_y < 0 :
  Bearing = 2PI - adj

If the target is due north, east, south or west of the aircraft the bearing is 0, PI/2, PI or 3PI/4 respectively.

2.1.20.3.10.5 External Interfaces

Called by:
PLATFORM_MOVEMENT_MODEL.UPDATE_PLATFORM_POSITIONS

Calls:
None.

2.1.20.3.10.6 Special Considerations

A target 0 must always exist.

2.1.21 Package *Plot File* (Plot_File_Pkg)

2.1.21.1 Package Description

This package contains utilities to manipulate the Plot file.

2.1.21.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:
subtype Number_Type
type Target_End_Record_Type

Data Declarations:

```
Plot_LFN              : Text_io.file_type;
Targ_LFN              : Text_io.file_type;
Plot_file_exception   : exception;
```

```
Quote              : String (1..1);
Degprad            : constant := 180.0/3.1415927;
NMprad             : constant := 10800.0/3.1415927;
Init_flag          : boolean;
Orgx, Orgy         : short_float;
Cos_Lat            : short_float;
Xlo,  Ylo          : short_float;
Xhi,  Yhi          : short_float;
End_Records        : array (Number_Type) of Target_End_Record_Type;
```

2.1.21.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Close_Plot_File
Open_Plot_File
Write_Plot_File

2.1.21.3.1 Unit *Close_Plot_File*

2.1.21.3.1.1 Purpose

This function closes Plot_file.

2.1.21.3.1.2 Input/Output Parameters

None.

2.1.21.3.1.3 Local Data

```
Cen          : short_float;
DX           : short_float;
DY           : short_float;
Half_Diff    : short_float;
Scale_Strx   : string (1 .. 20);
Scale_Stry   : string (1 .. 20);
```

2.1.21.3.1.4 Processing

Compute min and max statistics records to the plot file.

Write a command to raise the pen for all plot items.

Write a command to represent the minimums and maximums for scaling purposes.

Close the plot file.

2.1.21.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None

2.1.21.3.1.6 Special Considerations

None.

2.1.21.3.2 Unit *Open_Plot_File*

2.1.21.3.2.1 Purpose

This function opens Plot_file as text file for writing.

2.1.21.3.2.2 Input/Output Parameters

None.

2.1.21.3.2.3 Local Data

None.

2.1.21.3.2.4 Processing

Create/open a file named targ_file.dat for use in recording plotable data.

2.1.21.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None

2.1.21.3.2.6 Special Considerations

None.

2.1.21.3.3 Unit *Write_Plot_File*

2.1.21.3.3.1 Purpose

This function writes a line to plot_file.dat and targ_file.dat.

2.1.21.3.3.2 Input/Output Parameters

```
Number   : in integer;
Time     : in short_float;
Lat_Rad  : in short_float;
Lon_Rad  : in short_float;
```

2.1.21.3.3.3 Local Data

```
XX      : short_float;
YY      : short_float;
Strx    : string (1 .. 20);
Stry    : string (1 .. 20);
Strlat  : string (1 .. 20);
Strlon  : string (1 .. 20);
```

2.1.21.3.3.4 Processing

Convert latitude and longitude to plot position coordinates in nm north of the equator and east of the prime meridian.

Write plot position record to plot file identifying the target number.

Accumulate minimum and maximum statistics on position.

2.1.21.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None

2.1.21.3.3.6 Special Considerations

None.

2.1.22 Package *Scenario File Object* (Scenario_File_Obj_Pkg)

2.1.22.1 Package Description

This package provides for the creation of the Scenario File input to the Update IV testbed. It is invoked by ATSG Main and requests information from the Event Table Object, Platform Movement Model Object, ESM Table Object, Acoustic Data Object, and the Sensor Utilities.

2.1.22.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

```
type Object_Count_Type
type Update_Count_Type
```

Data Declarations:

```
AC_Position_Info            :
            SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
AC_Value                    :
            DBDTAC_AIRCRAFT_INCLUDE.DBDTAC_AIRCRAFT_REC_TYPE;
Add_Up_String               : string(1 .. 5);
Bearing                     :
            SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_Bearing_TYPE;
Comment_Flag                : boolean;
Contact_Number              :
     DBDTZD_Contact_Number_Type_Include.DBDTZD_Contact_Number_Type;
End_Of_Events               : boolean;
ESM_Data                    :  ESM_Table_Obj_Pkg.Emitter_Event_Type;
ESM_Data_Rec                :
     CRCRGLCR_Correlation_Type_Include.CRCRGLCR_Classification_Record_Type;
Event                       : Event_Table_Obj_Pkg.Event_File_Type;
```

```
Free_Form_Value             :
          DBDTBM_FREE_FORM_INCLUDE.DBDTBM_FREE_FORM_REC_TYPE;
Ground_Range                :
          SYGLME_MEASURE_TYPE_INCLUDE.SYGLME_NMILES_TYPE;
Half_Deltime                :  ATSG_Gen_Pkg.Event_Time_Type;
IRDS_Counts                 : Update_Count_Type;
IRDS_Flag                   : boolean;
IRDS_Object_Counts          : Object_Count_Type;
IRDS_Value                  :
          DBDTEU_Track_IRDS_Include.DBDTEU_Track_IRDS_REC_TYPE;
Itime                       : Integer;
Last_AC_Update_Time         :  ATSG_Gen_Pkg.Event_Time_Type;
Line_Count                  : integer;
MAD_Counts                  : Update_Count_Type;
MAD_Flag                    : boolean;
MAD_Object_Counts           : Object_Count_Type;
MAD_Track                   :
               DBDTEV_Track_MAD_Include.DBDTEV_Track_MAD_Rec_Type;
Man_Radar_Counts            : Update_Count_Type;
Man_Radar_Flag              : boolean;
Man_Radar_Object_Counts     : Object_Count_Type;
Man_Radar_Value             :
   DBDTEX_Track_Manual_Radar_Include.DBDTEX_Track_Manual_Radar_Rec_Type;
MOT_Value                   :
     DBDTDE_MARK_ON_TOP_INCLUDE.DBDTDE_MARK_ON_TOP_REC_TYPE ;
Plot_Time                   :  ATSG_Gen_Pkg.Event_Time_Type;
PMM_POSITION_INFO           :
               PLATFORM_MOVEMENT_MODEL.POSITION_INFO_TYPE;
Quote                       : string (1 .. 1);
Slant_Range                 :
               SYGLME_MEASURE_TYPE_INCLUDE.SYGLME_NMILES_TYPE;
Suffix                      : Pack_Strings_Pkg.String_Six_Type;
Suffixq                     : Pack_Strings_Pkg.String_7_Type;
Tgt_Position_Info           :
          SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
True_Tgt_Pos_Info           :
          SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
```

```
TWS_Radar_Counts          : Update_Count_Type;
TWS_Radar_Flag            : boolean;
TWS_Radar_Object_Counts   : Object_Count_Type;
TWS_Radar_Value           :
     DBDTEY_Track_TWS_Radar_Include.DBDTEY_Track_TWS_Radar_REC_TYPE;
Update_Object_ID          : ATSG_Gen_Pkg.Object_ID_Type;
Visual_Counts             : Update_Count_Type;
Visual_Flag               : boolean;
Visual_Object_Counts      : Object_Count_Type;
Visual_Value              :
     DBDTEZ_TRACK_VISUAL_INCLUDE.DBDTEZ_TRACK_VISUAL_REC_TYPE;
Zero_Flag                 : boolean;
```

2.1.22.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Generate_Scenario_File

2.1.22.3.1 Unit *Generate_Scenario_File*

2.1.22.3.1.1 Purpose

This Module is invoked by ATSG Main to control the acquisition of the Event data and to control the building of the Scenario file. This module invokes the EVENT_TABLE first to get the next event to be processed. If the event is a detection type event Generate_Scenario_File then invokes the Platform Movement Model Object to update the platform positions and provide the needed positional data for the event. If the detection event is an ESM or ACOUSTIC Detection, the ESM Table Object or the Acoustic Table object is invoked to provide the necessary sensor related data. Then the Sensor Utility for the appropriate sensor detection type is called to allow for the maintenance of the filters and direction cosine matrices used by the tracking software. If it is a non-detection type event, the appropriate comments are appended to the Scenario File to instruct the testbed operator of the appropriate input or responses.

2.1.22.3.1.2 Input/Output Parameters

None.

2.1.22.3.1.3 Local Data

None.

2.1.22.3.1.4 Processing

Generate a compilable scenario file by processing the entire event table.
For each event, determine platform positions and write the associated data object to the declaration scenario file and the appropriate command or comment to the command scenario file.
Append the declaration file and the command file to build the scenario file.

2.1.22.3.1.5 External Interfaces

Called by:

atsg_main

Calls:

AC_State_Pkg.Init_AC_State
AC_State_Pkg.Write_AC_State
ATSGSUAC_AIRCRAFT_PKG.ATSGSUAC_RECEIVE_AIRCRAFT_DATA
ATSGSUFF_FREE_FORM_PKG.ATSGSUFF_RECEIVE_FREE_FORM_DATA
ATSGSUIM_IRDS_Sensor_Pkg.ATSGSUIM_Receive_IRDS_Data
ATSGSUMM_MAD_Sensor_Pkg.ATSGSUMM_Receive_MAD_Data
ATSGSURM_Manual_Radar_Sensor_Pkg.ATSGSURM_Receive_Manual_Radar_Data
ATSGSUTW_TWS_RADAR_Sensor_Pkg.ATSGSUTW_Receive_TWS_RADAR_Data
ATSGSUVC_VISUAL_CONTACT_PKG.
ATSGSUVC_RECEIVE_VISUAL_CONTACT_DATA
ESM_Table_Obj_Pkg.Get_Data
Event_Table_Obj_Pkg.Get_Next_Event
Init_SF_Pkg.Write_General_Types
Init_SF_Pkg.Write_General_Withs
IRDS_Object_Out_Pkg.IRDS_Add_Header
IRDS_Object_Out_Pkg.IRDS_Out
IRDS_Object_Out_Pkg.IRDS_Upd_Header
MAD_Object_Out_Pkg.MAD_Add_Header
MAD_Object_Out_Pkg.MAD_Out
MAD_Object_Out_Pkg.MAD_Upd_Header
Man_Radar_Object_Out_Pkg.Man_Radar_Add_Header

Man_Radar_Object_Out_Pkg.Man_Radar_Out
Man_Radar_Object_Out_Pkg.Man_Radar_Upd_Header
Mark_Time_Object_Out_Pkg.Mark_Time_OUT
MOT_Object_Out_Pkg.MOT_OUT
Object_Manager_Pkg.Get_Operator_Action_Characteristics
Object_Manager_Pkg.Get_Operator_Response_Characteristics
Pack_Strings_Pkg.Build_Suffix
Pack_Strings_Pkg.Pack_Strings
Pack_Strings_Pkg.Quote_Terminate
PLATFORM_MOVEMENT_MODEL.UPDATE_PLATFORM_POSITIONS
Plot_File_Pkg.Close_Plot_File
Plot_File_Pkg.Open_Plot_File
Plot_File_Pkg.Write_Plot_File
Schedule_Pkg.Schedule
TWS_RADAR_Object_Out_Pkg.TWS_RADAR_Add_Header
TWS_RADAR_Object_Out_Pkg.TWS_RADAR_Out
TWS_RADAR_Object_Out_Pkg.TWS_RADAR_Upd_Header
Visual_Object_Out_Pkg.Visual_Add_Header
Visual_Object_Out_Pkg.Visual_Out
Visual_Object_Out_Pkg.Visual_Upd_Header
Warn_Pkg.Warn_User
Write_Files_Pkg.Close_Scenario_File
Write_Files_Pkg.Copy_Command_File
Write_Files_Pkg.Copy_Declare_File
Write_Files_Pkg.Open_Command_File
Write_Files_Pkg.Open_Declare_File
Write_Files_Pkg.Open_Scenario_File
Write_Files_Pkg.Write_Scenario_File
Write_Files_Pkg.Write_Scenario_File_WU
Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Declare_File

2.1.22.3.1.6 Special Considerations

None.

2.1.23 Package *Schedule* (Schedule_Pkg)

2.1.23.1 Package Description

This package places the SCHEDULE command on the scenario file.

2.1.23.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

```
Count        : integer;
```

2.1.23.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Schedule

2.1.23.3.1 Unit *Schedule*

2.1.23.3.1.1 Purpose

This procedure writes a SCHEDULE line followed by a WAIT command.

2.1.23.3.1.2 Input/Output Parameters

```
Name        : in string;
MS_Delay    : in integer;
Line_Count  : in out integer
```

2.1.23.3.1.3 Local Data

None.

2.1.23.3.1.4 Processing

Write a SCHEDULE line followed by a WAIT line to the command scenario file.

2.1.23.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Pack_Strings_Pkg.Pack_Strings
Write_Files_Pkg.Write_To_Command_File
Write_Files_Pkg.Write_To_Command_File

2.1.23.3.1.6 Special Considerations

None.

2.1.24 Package *Track Data Object* (Track_Data_Out_Pkg)

2.1.24.1 Package Description

This package writes Track data to the scenario file.

2.1.24.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

None.

2.1.24.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Write_Track_Data

2.1.24.3.1 Unit *Write_Track_Data*

2.1.24.3.1.1 Purpose

This procedure writes Track Data to the scenario file.

2.1.24.3.1.2 Input/Output Parameters

```
Track_Data  :    in CRCRGLCR_CORRELATION_TYPE_INCLUDE.
                            CRCRGLCR_TRACK_RECORD_TYPE
```

2.1.24.3.1.3 Local Data

```
Str20       : string (1 .. 20);
```

2.1.24.3.1.4 Processing

Write the TRACK_DATA component of a tactical object to the declaration scenario file.

2.1.24.3.1.5 External Interfaces

Called by:

Man_Radar_Object_Out_Pkg.Man_Radar_Out
TWS_Radar_Object_Out_Pkg.TWS_Radar_Out
Visual_Object_Out_Pkg.Visual_Out

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.24.3.1.6 Special Considerations

None.

2.1.25 Package *TWS Radar* (TWS_Radar_Object_Out_Pkg)

2.1.25.1 Package Description

This package writes the TWS Radar data to the scenario file.

2.1.25.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:
None.

Data Declarations:
Quote : string (1..1);

2.1.25.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

TWS_Radar_Add_Header
TWS_Radar_Out
TWS_Radar_UPD_Header

2.1.25.3.1 Unit *TWS_Radar_Add_Header*

2.1.25.3.1.1 Purpose

This procedure writes the TWS Radar add header to the scenario file.

2.1.25.3.1.2 Input/Output Parameters

None.

2.1.25.3.1.3 Local Data

None.

2.1.25.3.1.4 Processing

Write the header portion of the TWS Radar object which is specific to the ADD tactical TWS Radar object.

2.1.25.3.1.5 External Interfaces

Called by:
Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:
Write_Files_Pkg.Write_To_Declare_File

2.1.25.3.1.6 Special Considerations

None.

2.1.25.3.2 Unit *TWS_Radar_Out*

2.1.25.3.2.1 Purpose

This procedure writes the TWS Radar object value to the scenario file.

2.1.25.3.2.2 Input/Output Parameters

```
TWS_Radar_Value      : DBDTEY_TRACK_TWS_Radar_INCLUDE.
                      DBDTEY_TRACK_TWS_Radar_REC_TYPE
```

2.1.25.3.2.3 Local Data

```
Complete_Line   : boolean;
Str20           : string (1 .. 20);
```

2.1.25.3.2.4 Processing

Write the body portion of the TWS Radar object which is common to both ADD and UPDATE tactical TWS Radar object.

2.1.25.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File
Track_Data_Out_Pkg.Write_Track_Data

2.1.25.3.2.6 Special Considerations

None.

2.1.25.3.3 Unit *TWS_Radar_UPD_Header*

2.1.25.3.3.1 Purpose

This procedure writes the TWS Radar UPD header to the scenario file.

2.1.25.3.3.2 Input/Output Parameters

ID : ATSG_Gen_Pkg.Object_ID_Type

2.1.25.3.3.3 Local Data

None.

2.1.25.3.3.4 Processing

Write the header portion of the TWS Radar object which is specific to the UPDATE tactical TWS Radar object.

2.1.25.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.25.3.3.6 Special Considerations

None.

2.1.26 Package *Math Utilities* (UT_Math_Pkg)

2.1.26.1 Package Description

This package supplements UTMAMA_MATH_UTILS_PKG.

2.1.26.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

None.

2.1.26.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Cartesian_To_Polar
Polar_To_Cartesian

2.1.26.3.1 Unit *Cartesian_To_Polar*

2.1.26.3.1.1 Purpose

This procedure converts Cartesian coordinates to polar coordinates.

2.1.26.3.1.2 Input/Output Parameters

Delta_X	: in float;
Delta_Y	: in float;
Delta_R	: out float;
Bearing	: out float;

2.1.26.3.1.3 Local Data

None.

2.1.26.3.1.4 Processing

Convert Cartesian coordinates to Polar.

2.1.26.3.1.5 External Interfaces

Called by:

None.

Calls:

UTMAMA_MATH_UTILS_PKG.sqrt
UTMAMA_MATH_UTILS_PKG.UTMAMA_ARCTAN2_FUNC
UTMAMA_MATH_UTILS_PKG.UTMAMA_FLOAT
UTMAMA_MATH_UTILS_PKG.UTMAMA_NATURAL_FLOAT

2.1.26.3.1.6 Special Considerations

None.

2.1.26.3.2 Unit *Polar_To_Cartesian*

2.1.26.3.2.1 Purpose

This procedure converts Polar coordinates to Cartesian coordinates.

2.1.26.3.2.2 Input/Output Parameters

Delta_R	: in float;
Bearing	: in float;
Delta_X	: out float;
Delta_Y	: out float;

2.1.26.3.2.3 Local Data

None.

2.1.26.3.2.4 Processing

Convert Polar coordinates to Cartesian.

2.1.26.3.2.5 External Interfaces

Called by:

None.

Calls:

UTMAMA_MATH_UTILS_PKG.cos
UTMAMA_MATH_UTILS_PKG.sin

2.1.26.3.2.6 Special Considerations

None.

2.1.27 Package *Visual Object* (Visual_Object_Out_Pkg)

2.1.27.1 Package Description

This package writes the Visual data to the scenario file.

2.1.27.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Quote : string (1..1);

2.1.27.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Visual_Add_Header
Visual_Out
Visual_UPD_Header

2.1.27.3.1 Unit *Visual_Add_Header*

2.1.27.3.1.1 Purpose

This procedure writes the Visual add header to the scenario file.

2.1.27.3.1.2 Input/Output Parameters

None.

2.1.27.3.1.3 Local Data

None.

2.1.27.3.1.4 Processing

Write the header portion of the Visual object which is specific to the ADD tactical Visual object.

2.1.27.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.27.3.1.6 Special Considerations

None.

2.1.27.3.2 Unit *Visual_Out*

2.1.27.3.2.1 Purpose

This procedure writes the Visual object value to the scenario file.

2.1.27.3.2.2 Input/Output Parameters

```
Visual_Value    : in DBDTEZ_TRACK_VISUAL_INCLUDE.
                              DBDTEZ_TRACK_VISUAL_REC_TYPE
```

2.1.27.3.2.3 Local Data

```
Complete_Line   : boolean;
Str20           : string (1 .. 20);
```

2.1.27.3.2.4 Processing

Write the body portion of the Visual object which is common to both ADD and UPDATE tactical Visual object.

2.1.27.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Track_Data_Out_Pkg.Write_Track_Data

Write_Files_Pkg.Write_To_Declare_File

2.1.27.3.2.6 Special Considerations

None.

2.1.27.3.3 Unit *Visual_UPD_Header*

2.1.27.3.3.1 Purpose

This procedure writes the Visual UPD header to the scenario file.

2.1.27.3.3.2 Input/Output Parameters

ID : ATSG_Gen_Pkg.Object_ID_Type

2.1.27.3.3.3 Local Data

None.

2.1.27.3.3.4 Processing

Write the header portion of the Visual object which is specific to the UPDATE tactical Visual object.

2.1.27.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_To_Declare_File

2.1.27.3.3.6 Special Considerations

None.

2.1.28 Package *Warn* (Warn_Pkg)

2.1.28.1 Package Description

This package contains a procedure to wars the user of an error by an alert and a beep.

2.1.28.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:
None.

2.1.28.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Warn_User

2.1.28.3.1 Unit *Warn_User*

2.1.28.3.1.1 Purpose

This procedure warns the user of an error by an alert and a beep.

2.1.28.3.1.2 Input/Output Parameters

None.

2.1.28.3.1.3 Local Data

dumchar : character;

2.1.28.3.1.4 Processing

Print an error message and sound a BEEP.

2.1.28.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.28.3.1.6 Special Considerations

None.

2.1.29 Package *Write* (Write_Files_Pkg)

2.1.29.1 Package Description

This package contains procedures to manipulate the Scenario File, the Command File, and the Declare File

2.1.29.2 Package Data

The following data is defined in this package. More information for this data can be found in the Data Dictionary.

Type Declarations:

None.

Data Declarations:

Command_File	: Text_io.file_type;
Command_File_Exception	: exception;
Declare_File	: Text_io.file_type;
Declare_File_Exception	: exception;
LINESIZE	: constant := 79;
Scenario_file_exception	: exception;
Scenario_LFN	: Text_io.file_type;

2.1.29.3 Package Units

The following is a list of all the units in this package. Each unit is described in detail in the following sections.

Open_Scenario_File
Close_Scenario_File
Write_Scenario_File
Write_Scenario_File_WU
Open_Command_File
Write_to_Command_File
Copy_Command_File
Open_Declare_File
Write_To_Declare_File
Copy_Declare_File

2.1.29.3.1 Unit *Open_Scenario_File*

2.1.29.3.1.1 Purpose

This procedure opens the Scenario file as text file for writing.

2.1.29.3.1.2 Input/Output Parameters

None.

2.1.29.3.1.3 Local Data

None.

2.1.29.3.1.4 Processing

Create/open file named scenario_file.a as text file for writing.

2.1.29.3.1.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.1.6 Special Considerations

None.

2.1.29.3.2 Unit *Close_Scenario_File*

2.1.29.3.2.1 Purpose

This procedure closes the Scenario File.

2.1.29.3.2.2 Input/Output Parameters

None.

2.1.29.3.2.3 Local Data

None.

2.1.29.3.2.4 Processing
Close the scenario file and its declaration and command files.

2.1.29.3.2.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.2.6 Special Considerations
None.

2.1.29.3.3 Unit *Write_Scenario_File*

2.1.29.3.3.1 Purpose
This procedure writes a line to the Scenario file.

2.1.29.3.3.2 Input/Output Parameters

line : in String

2.1.29.3.3.3 Local Data
None.

2.1.29.3.3.4 Processing
Write a line of text to the scenario file terminating with EOL.

2.1.29.3.3.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File
Write_Files_Pkg.Write_Scenario_File_WU

Calls:

None.

2.1.29.3.3.6 Special Considerations

None.

2.1.29.3.4 Unit *Write_Scenario_File_WU*

2.1.29.3.4.1 Purpose

This procedure writes "with" and "use" lines to the scenario file for Package_Name.

2.1.29.3.4.2 Input/Output Parameters

Package_Name : in String

2.1.29.3.4.3 Local Data

None.

2.1.29.3.4.4 Processing

Write with and use clauses to the scenario file for the input package name.

2.1.29.3.4.5 External Interfaces

Called by:

Init_SF_Pkg.Write_General_Withs
Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

Write_Files_Pkg.Write_Scenario_File

2.1.29.3.4.6 Special Considerations

None.

2.1.29.3.5 Unit *Open_Command_File*

2.1.29.3.5.1 Purpose

This procedure opens the command file as text file for writing.

2.1.29.3.5.2 Input/Output Parameters

None.

2.1.29.3.5.3 Local Data

None.

2.1.29.3.5.4 Processing

Create/open file named command_file.temp as text file for writing the temporary scenario file commands.

2.1.29.3.5.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.5.6 Special Considerations

None.

2.1.29.3.6 Unit *Write_to_Command_File*

2.1.29.3.6.1 Purpose

This procedure writes a line to the command_file.temp.

2.1.29.3.6.2 Input/Output Parameters

EOL : in boolean;
line : in String;

2.1.29.3.6.3 Local Data

None.

2.1.29.3.6.4 Processing

Write a line of text to the scenario command file terminating with an EOL if EOL flag is true.

2.1.29.3.6.5 External Interfaces

Called by:

AC_State_Pkg.Write_AC_State
Free_Form_Obj_Pkg.Write_Free_Form_Obj
Init_SF_Pkg.Write_CR_Init
Init_SF_Pkg.Write_Tac_Init
Mark_Time_Object_Out_Pkg.Mark_Time_OUT
MOT_Object_Out_Pkg.MOT_OUT
Scenario_File_Obj_Pkg.Generate_Scenario_File
Schedule_Pkg.Schedule
Schedule_Pkg.Schedule

Calls:

None.

2.1.29.3.6.6 Special Considerations

None.

2.1.29.3.7 Unit *Copy_Command_File*

2.1.29.3.7.1 Purpose

This unit appends the command scenario file to the scenario file.

2.1.29.3.7.2 Input/Output Parameters

None.

2.1.29.3.7.3 Local Data

One_Char : character;

2.1.29.3.7.4 Processing

Rewind and copy the command scenario file to the end of scenario file.

2.1.29.3.7.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.7.6 Special Considerations

None.

2.1.29.3.8 Unit *Open_Declare_File*

2.1.29.3.8.1 Purpose

This procedure opens the declaration file as a text file for writing.

2.1.29.3.8.2 Input/Output Parameters

None.

2.1.29.3.8.3 Local Data

None.

2.1.29.3.8.4 Processing

Create/open file named declare_file.temp as text file for writing the temporary scenario file declarations.

2.1.29.3.8.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.8.6 Special Considerations

None.

2.1.29.3.9 Unit *Write_To_Declare_File*

2.1.29.3.9.1 Purpose

This procedure writes to the declaration file.

2.1.29.3.9.2 Input/Output Parameters

EOL	: in boolean;
line	: in String;

2.1.29.3.9.3 Local Data

None.

2.1.29.3.9.4 Processing

Write a line of text to the scenario declaration file terminating with an EOL if EOL flag is true.

2.1.29.3.9.5 External Interfaces

Called by:

AC_State_Pkg.Init_AC_State
AC_State_Pkg.Write_AC_State
Free_Form_Obj_Pkg.Write_Free_Form_Obj
Init_SF_Pkg.Write_CR_Init
Init_SF_Pkg.Write_General_Types
Init_SF_Pkg.Write_Tac_Init
IRDS_Object_Out_Pkg.IRDS_Add_Header
IRDS_Object_Out_Pkg.IRDS_Out
IRDS_Object_Out_Pkg.IRDS_UPD_Header
MAD_Object_Out_Pkg.MAD_Add_Header
MAD_Object_Out_Pkg.MAD_OUT
MAD_Object_Out_Pkg.MAD_UPD_Header
Man_Radar_Object_Out_Pkg.Man_Radar_Add_Header
Man_Radar_Object_Out_Pkg.Man_Radar_Out
Man_Radar_Object_Out_Pkg.Man_Radar_UPD_Header
Mark_Time_Object_Out_Pkg.Mark_Time_Init
Mark_Time_Object_Out_Pkg.Mark_Time_OUT
Measurement_Data_Out_Pkg.Write_Measurement_Data
MOT_Object_Out_Pkg.MOT_Init
MOT_Object_Out_Pkg.MOT_OUT
Scenario_File_Obj_Pkg.Generate_Scenario_File
Track_Data_Out_Pkg.Write_Track_Data
TWS_Radar_Object_Out_Pkg.TWS_Radar_Add_Header

TWS_Radar_Object_Out_Pkg.TWS_Radar_Out
TWS_Radar_Object_Out_Pkg.TWS_Radar_UPD_Header
Visual_Object_Out_Pkg.Visual_Add_Header
Visual_Object_Out_Pkg.Visual_Out
Visual_Object_Out_Pkg.Visual_UPD_Header
Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.9.6 Special Considerations

None.

2.1.29.3.10 Unit *Copy_Declare_File*

2.1.29.3.10.1 Purpose

This procedure copies all of declare_file.temp to Scenario_LFN file.

2.1.29.3.10.2 Input/Output Parameters

None.

2.1.29.3.10.3 Local Data

One_Char : character;

2.1.29.3.10.4 Processing

Rewind and copy the declare scenario file to the end of scenario file.

2.1.29.3.10.5 External Interfaces

Called by:

Scenario_File_Obj_Pkg.Generate_Scenario_File

Calls:

None.

2.1.29.3.10.6 Special Considerations

None.

AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
DESIGN DESCRIPTION

APPENDIX A: DATA DICTIONARY

12 June 1992

AC_Data : in PLATFORM_MOVEMENT_MODEL.POSITION_INFO_TYPE;
Description: Data returned from the platform package for the A/C.
Package: AC_State_Pkg.
Unit: Write_AC_State.
Usage: Formal Parameter.

AC_DC_RATE : in FLOAT;
Description: Acceleration / Deceleration Rate.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

AC_DC_Rate : float;
Description: Acceleration/decelleration of target or A/C in ft/sec^2.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

AC_Position_Info :
SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
Description: Target pos data fm PMM to Scenario_Obj for sensor packages.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

AC_TO_TARGET_BEARING : out FLOAT;
Description: Aircraft to target bearing.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

AC_TO_TARGET_FLAT_RNG : out FLOAT;
Description: Aircraft to target flat range.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

AC_TO_TARGET_SLANT_RNG : FLOAT;
Description: The AC to target slant range.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

AC_TO_TARGET_SLANT_RNG : out FLOAT;
Description: The AC to target slant range.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

AC_Value : DBDTAC_AIRCRAFT_INCLUDE.DBDTAC_AIRCRAFT_REC_TYPE;
Description: Value returned from the A/C utility for the value of the scenario file A/C object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

AC_Value : in DBDTAC_AIRCRAFT_INCLUDE.DBDTAC_AIRCRAFT_REC_TYPE;
Description: Value returned from the A/C utility for the value of the scenario file A/C object.
Package: AC_Obj_Pkg.
Unit: Write_AC_Obj.
Usage: Formal Parameter.

ACOUSTIC_SS : float;
Description: perturbation in ft.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Action : Operator_Pkg.Action_Type;
Description: The operator action selection being read and processed.
Package: Input_File_Obj_Pkg.
Unit: Read_DOIT_Event.
Usage: Local Data.

Action : Operator_Pkg.Action_Type;
Description: The operator action selection being read and processed.
Package: Object_Manager_Pkg.
Unit: Get_Operator_Action_Characteristics.
Usage: Formal Paramater.

Action : Operator_Pkg.Situation_Response_Type;
Description: The operator response being read and processed.
Package: Input_File_Obj_Pkg.
Unit: Read_DSRT_Event.
Usage: Local Data.

Action_Object_ID_Array : Action_Object_ID_Array_Type := (others => 0);
Description: An indication of how many additional objects are generated by the actions.
Package: Object_Manager_Pkg spec.
Usage: ATSG Global Data.

Add_Up_String : string(1 .. 5);
Description: The string containing the name of either the update or add task number.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

ADJ : FLOAT;
Description: The value of the angle adjacent to the hypotenuse.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Local Data.

Aircraft : constant INTEGER;
Description: Index value of the P3 Aircraft Platform.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

ALFA : FLOAT;
Description: An intermediate angle calculation.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Local Data.

Altitude_Depth : float;
Description: Latest target position for Platform_Movement_Model.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

ATSG_Input_Value_Error : exception;
Description: The value of an ATSG input is invalid. (Use Dlevel > 0 to diagnose.)
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

AVE_SPEED : FLOAT;
Description: The average speed of this line update.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Local Data.

Bearing : in float;
Description: Bearing in radians from X axis toward Y axis from 0 to 2 PI.
Package: UT_Math_Pkg.
Unit: Polar_To_Cartesian.
Usage: Formal Parameter.

Bearing : out float;
Description: Bearing in radians from X axis toward Y axis from 0 to 2 PI.
Package: UT_Math_Pkg.
Unit: Cartesian_To_Polar.
Usage: Formal Parameter.

Bearing : SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_Bearing_TYPE;
Description: True bearing A/C to target from PMM to Scenario_Obj for sensor packages (radians).
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

BETA : FLOAT;
Description: An intermediate angle calculation.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Local Data.

Blank_NATO : Platform_Type := "   ";
Description: A value of null for the NATO class.
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data.

C1 : constant FLOAT := 0.008
Description: Unclassified angular cutoff from VCCLCO_VC_CLASSIFIED_CONSTANTS_INCLUDE.VCCLCO_IRDS_BEARING_CONSTANT_C1_C;
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

C1 : FLOAT;
Description: Intermediate value using rotation matrix C.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

C2 : FLOAT;
Description: Intermediate value using rotation matrix C.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

C2_over_C1 : constant FLOAT := 0.1 / C1

Description: Unclassified angular standard deviation using VCCLCO_VC_CLASSIFIED _CONSTANTS_INCLUDE.VCCLCO_IRDS_BEARING_CONSTANT_C2_C / C1.

Unit: COMPUTE_DEFAULT_OFFSET.

Usage: Local Data.

C3 : FLOAT;

Description: Intermediate value using rotation matrix C.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: UPDATE_LAT_LON.

Usage: Local Data.

C_INV_1_1 : FLOAT;

Description: Intermediate matrix inverse result.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: ESTABLISH_INITIAL_LAT_LON.

Usage: Local Data.

C_INV_1_2 : FLOAT;

Description: Intermediate matrix inverse result.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: ESTABLISH_INITIAL_LAT_LON.

Usage: Local Data.

C_INV_1_3 : FLOAT;

Description: Intermediate matrix inverse result.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: ESTABLISH_INITIAL_LAT_LON.

Usage: Local Data.

C_INV_2_1 : FLOAT;

Description: Intermediate matrix inverse result.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: ESTABLISH_INITIAL_LAT_LON.

Usage: Local Data.

C_INV_2_2 : FLOAT;
Description: Intermediate matrix inverse result.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Local Data.

C_INV_2_3 : FLOAT;
Description: Intermediate matrix inverse result.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Local Data.

Cen : short_float;
Description: Center point for X and Y.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

COMM_SS : float;
Description: perturbation in ft.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Command_File : Text_io.file_type;
Description: Scenario command file local file name.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

Command_File_Exception : exception;
Description: An error in writing the scenario command file.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

COMMAND_PTR : COMMAND_PTR_TYPE;
Description: Command pointers point to the start and end segment to be executed by each platform. They are initialized to 1 and updated by MOVE_PLATFORM.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

COMMAND_PTR : in out COMMAND_PTR_TYPE;
Description: Command pointers point to the start and end segment to be executed by each platform. They are initialized to 1 and updated by MOVE_PLATFORM.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Formal Parameter.

Comment_Flag : boolean;
Description: Indicates whether a comment is being written to the sscnario_file.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Complete_Line : boolean;
Description: Indicates whether the text is to be terminated with EOL.
Package: IRDS_Object_Out_Pkg.
Unit: IRDS_Out.
Usage: Local Data.

Complete_Line : boolean;
Description: Indicates whether the text is to be terminated with EOL.
Package: MAD_Object_Out_Pkg.
Unit: MAD_Out.
Usage: Local Data.

Complete_Line : boolean;
Description: Indicates whether the text is to be terminated with EOL.
Package: Man_Radar_Object_Out_Pkg.
Unit: Manual_Radar_Out.
Usage: Local Data.

Complete_Line : boolean;
Description: Indicates whether the text is to be terminated with EOL.
Package: TWS_Radar_Object_Out_Pkg.
Unit: TWS_Radar_Out.
Usage: Local Data.

Complete_Line : boolean;
Description: Indicates whether the text is to be terminated with EOL.
Package: Visual_Object_Out_Pkg.
Unit: Visual_Out.
Usage: Local Data.

Constant_A_C : constant FLOAT := 110.0182 (yards);
Description: Visual Contact perturbation constant. Taken from VC_MARK_ON_TOP_CONTACT_TASK from atsgsuvc_visual_contact_body.a
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Constant_B_C : constant FLOAT := -0.034965;
Description: Visual Contact perturbation constant. Taken from VC_MARK_ON_TOP _CONTACT_TASK from atsgsuvc_visual_contact_body.a
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Constant_C_C : constant FLOAT := 0.000025525;
Description: Visual Contact perturbation constant. Taken form VC_MARK_ON_TOP _CONTACT_TASK from atsgsuvc_visual_contact_body.a
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Contact_Number : DBDTZD_Contact_Number_Type_Include.
DBDTZD_Contact_Number_Type;
Description: CONTACT Number of the MAD contact/track.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Cos_Lat : short_float;
Description: The cosine of the latitude of the initial point to be plotted.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Count : integer;
Description: The number of times this routine has been called for the purpose of uniqluy identifying the object being written.
Package: AC_State_Pkg body.
Usage: Package Global Data.

Count : integer;
Description: The number of times this routine has been called.
Package: AC_Obj_Pkg body.
Usage: Package Global Data.

Count : integer;
Description: The number of times this routine has been called.
Package: Mark_Time_Object_Out_Pkg body.
Usage: Package Global Data.

Count : integer;
Description: The number of times this routine has been called.
Package: MOT_Object_Out_Pkg body.
Usage: Package Global Data.

Count : integer;
Description: The number of times this routine has been called.
Package: Schedule_Pkg body.
Usage: Package Global Data.

Data_OK : boolean;
Description: Indicates input data is present.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Declare_File : Text_io.file_type;
Description: Scenario declare file local file name.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

Declare_File_Exception : exception;
Description: An error in writing the scenario declare file.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

Degprad : constant := 180.0/3.1415927;
Description: The number of angular degrees in a radian.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Degrees_to_Radians : constant FLOAT := PI / 180.0;
Description: PI [rad] / 180 [degrees]
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

DELTA_ANGLE : FLOAT;
Description: Change in heading between the beginning and end arc heading.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

Delta_R : in float;
Description: Hypotenuse of triangle with legs Delta_X and Delta_Y.
Package: UT_Math_Pkg.
Unit: Polar_To_Cartesian.
Usage: Formal Parameter.

Delta_R : out float;
Description: Hypotenuse of triangle with legs Delta_X and Delta_Y.
Package: UT_Math_Pkg.
Unit: Cartesian_To_Polar.
Usage: Formal Parameter.

DELTA_TIME : ATSG_Gen_Pkg.Event_Time_Type;
Description: The length of time for this update (sec).
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Local Data.

DELTA_X : FLOAT;
Description: The East-West distance from aircraft to target.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

DELTA_X : FLOAT;
Description: The East-West distance from aircraft to target.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Local Data.

Delta_X : in float;
Description: Cartesian coordinates to be converted.
Package: UT_Math_Pkg.
Unit: Cartesian_To_Polar.
Usage: Formal Parameter.

Delta_X : out float;
Description: Cartesian coordinates to be converted.
Package: UT_Math_Pkg.
Unit: Polar_To_Cartesian.
Usage: Formal Parameter.

DELTA_Y : FLOAT;
Description: The North-South distance from aircraft to target.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

DELTA_Y : FLOAT;
Description: The North-South distance from aircraft to target.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Local Data.

Delta_Y : in float;
Description: Cartesian coordinates to be converted.
Package: UT_Math_Pkg.
Unit: Cartesian_To_Polar.
Usage: Formal Parameter.

Delta_Y : out float;
Description: Cartesian coordinates to be converted.
Package: UT_Math_Pkg.
Unit: Polar_To_Cartesian.
Usage: Formal Parameter.

DELTA_Z_SQD : FLOAT;
Description: The Altitude distance from aircraft to target squared.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

DELTA_Z_SQD : FLOAT;
Description: The Altitude distance from aircraft to target squared.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Local Data.

DEP_ANGLE : FLOAT;
Description: The AC to target Depression angle.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

DIFF_X : FLOAT;
Description: Intermediate matrix difference result.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Local Data.

DIFF_Y : FLOAT;
Description: Intermediate matrix difference result.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Local Data.

DIFF_Z : FLOAT;
Description: Intermediate matrix difference result.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Local Data.

Direction : Platform_Movement_Model.Direction_Type;
Description: The direction of turn for a platform (RIGHT or LEFT).
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

DOIT_Event : D_OIT_Obj_Pkg.DOIT_Event_Type;
Description: Deterministic Operator Input events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

DOIT_Event : D_OIT_Obj_Pkg.DOIT_Event_Type;
Description: Tentative next DOIT event.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

DOIT_Event : in DOIT_Event_Type
Description: A record of DOIT operator action data.
Package: D_OIT_Obj_Pkg.
Unit: Store.
Usage: Formal Parameter.

DOIT_Event : out D_OIT_Obj_Pkg.DOIT_Event_Type
Description: Data to describe a DOIT event which is read.
Package: Input_File_Obj_Pkg.
Unit: Read_DOIT_Event.
Usage: Formal Parameter.

DOIT_Events : array (Event_Index_Type) of DOIT_Event_Type;
Description: Array of all DOIT events being used.
Package: D_OIT_Obj_Pkg body.
Usage: Package Global Data.

DSIT_Event : D_SIT_Obj_Pkg.DSIT_Event_Type;
Description: Deterministic Sensor Input events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

DSIT_Event : D_SIT_Obj_Pkg.DSIT_Event_Type;
Description: Tentative next DSIT event.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

DSIT_Event : in DSIT_Event_Type
Description: A record of DSIT data.
Package: D_SIT_Obj_Pkg.
Unit: Store.
Usage: Formal Parameter.

DSIT_Event : out D_SIT_Obj_Pkg.DSIT_Event_Type
Description: Data to describe a DSIT event.
Package: Input_File_Obj_Pkg.
Unit: Read_DSIT_Event.
Usage: Formal Parameter.

DSIT_Events : array (Event_Index_Type) of DSIT_Event_Type;
Description: Array of all DSIT events being used.
Package: D_SIT_Obj_Pkg body.
Usage: Package Global Data.

DSRT_Event : D_SRT_Obj_Pkg.DSRT_Event_Type;
Description: Deterministic Situation Response events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

DSRT_Event : D_SRT_Obj_Pkg.DSRT_Event_Type;
Description: Tentative next DSRT event.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

DSRT_Event : in DSRT_Event_Type
Description: A record of DSRT data.
Package: D_SRT_Obj_Pkg.
Unit: Store.
Usage: Formal Parameter.

DSRT_Event : out D_SRT_Obj_Pkg.DSRT_Event_Type
Description: Data to describe a DSRT event.
Package: Input_File_Obj_Pkg.
Unit: Read_DSRT_Event.
Usage: Formal Parameter.

DSRT_Events : array (Event_Index_Type) of DSRT_Event_Type;
Description: Array of all DSRT events being used.
Package: D_SRT_Obj_Pkg body.
Usage: Package Global Data.

Dumchar : character;
Description: A character buffer for scratch reading.
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Local Data.

dumchar : character;
Description: Any character entered to receive the operator response.
Package: Warn.
Unit: Warn_User.
Usage: Local Data.

Duration : float;
Description: The duration of the straight line segment of a PMM event in seconds.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

DX : short_float;
Description: The difference between max and minimum for X and Y.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

DY : short_float;
Description: The difference between max and minimum for X and Y.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

E_LAT : FLOAT;
Description: Pre-normalization Earth latitude of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

E_LON : FLOAT;
Description: Pre-normalization Earth longitude of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

Earth_Radius : constant FLOAT;
Description: Earths Radius.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

Emitter_Data : text_io.file_type;
Description: LFN for the emitter_data.dat.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Emitter_Event : ESM_Table_Obj_Pkg.Emitter_Event_Type;
Description: ESM events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Emitter_Event : in Emitter_Event_Type;
Description: A record listing the emitter and other characteristics of a target.
Package: ESM_Table_Obj_Pkg.
Unit: Store.
Usage: Formal Parameter.

Emitter_Event : out ESM_Table_Obj_Pkg.Emitter_Event_Type
Description: Data to describe an emitter event.
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Formal Parameter.

End_Of_Events : boolean;
Description: An indicator of no more events on the event table.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

End_Of_List : out boolean;
Description: Indicates that there are no more event of this type.
Package: D_OIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

End_Of_List : out boolean;
Description: Indicates that there are no more event of this type.
Package: D_SIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

End_Of_List : out boolean;
Description: Indicates that there are no more event of this type.
Package: D_SRT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

End_of_Lists : boolean;
Description: Indicates whether data is completely processed.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

End_Of_OIT : boolean;
Description: Indicates whether data is completely processed.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

End_Of_SIT : boolean;
Description: Indicates whether data is completely processed.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

End_Of_SRT : boolean;
Description: Indicates whether data is completely processed.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

End_Records : array (Number_Type) of Target_End_Record_Type;
Description: Latest plot position (x, y) of all targets in text.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

END_SPEED : FLOAT;
Description: The speed at the end of this line update.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Local Data.

END_TIME : ATSG_Gen_Pkg.Event_Time_Type;
Description: Begin_time + Duration of the segment.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

EOL : in boolean;
Description: Indicates whether a newline is to be placed at the end.
Package: Write_Files_Pkg.
Unit: Write_to_Command_File.
Usage: Formal Parameter.

EOL : in boolean;
Description: Indicates whether a newline is to be placed at the end.
Package: Write_Files_Pkg.
Unit: Write_To_Declare_File.
Usage: Formal Parameter.

ESM_Data : ESM_Table_Obj_Pkg.Emitter_Event_Type;
Description: Target/Emitter data returned from ESM_Table_Obj.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

ESM_Data : out Emitter_Event_Type;
Description: A record listing the emitter and other characteristics of a target.
Package: ESM_Table_Obj_Pkg.
Unit: Get_Data.
Usage: Formal Parameter.

ESM_Data_Rec : CRCRGLCR_Correlation_Type_Include.
CRCRGLCR_Classification_Record_Type;
Description: Platform description of the contact.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

ESM_SS : float;
Description: perturbation in ft.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

ET_Index : ATSG_Gen_Pkg.Event_Count_Type;
Description: Index to event table being built.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

Event : Event_Table_Obj_Pkg.Event_File_Type;
Description: The event found in the event table.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Event : in Event_Table_Obj_Pkg.Event_File_Type;
Description: Data returned from the Event package for the time.
Package: AC_State_Pkg.
Unit: Write_AC_State.
Usage: Formal Parameter.

Event : out DOIT_Event_Type
Description: The output D_OIT event (if not End_Of_List).
Package: D_OIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Event : out DSIT_Event_Type
Description: The output DSIT event (if not End_Of_List).
Package: D_SIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Event : out DSRT_Event_Type
Description: The output DSRT event (if not End_Of_List).
Package: D_SRT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Event : out Event_File_Type
Description: A record of the event to be processed into the scenario file.
Package: Event_Table_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Event_Data : text_io.file_type;
Description: LFN for the event_data.dat.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Event_Get_Count : Event_Count_Type;
Description: Count of DOIT event being extracted.
Package: D_OIT_Obj_Pkg body.
Usage: Package Global Data.

Event_Get_Count : Event_Count_Type;
Description: Count of DSIT event being extracted.
Package: D_SIT_Obj_Pkg body.
Usage: Package Global Data.

Event_Get_Count : Event_Count_Type;
Description: Count of DSRT event being extracted.
Package: D_SRT_Obj_Pkg body.
Usage: Package Global Data.

Event_Index : ATSG_Gen_Pkg.Event_Count_Type;
Description: Events counter for access order.
Package: Event_Table_Obj_Pkg body.
Usage: Package Global Data.

EVENT_KIND : in ATSG_Gen_Pkg.Event_Kind_TYPE;
Description: The sensor which precipitated this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Formal Parameter.

EVENT_KIND : in ATSG_Gen_Pkg.Event_Kind_TYPE;
Description: The sensor which precipitated this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Formal Parameter.

EVENT_KIND : in ATSG_Gen_Pkg.Event_Kind_Type;
Description: The sensor which precipitated this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Formal Parameter.

Event_Store_Count : Event_Count_Type;
Description: Count of DOIT event being stored.
Package: D_OIT_Obj_Pkg body.
Usage: Package Global Data.

Event_Store_Count : Event_Count_Type;
Description: Count of DSIT event being stored.
Package: D_SIT_Obj_Pkg body.
Usage: Package Global Data.

Event_Store_Count : Event_Count_Type;
Description: Count of DSRT event being stored.
Package: D_SRT_Obj_Pkg body.
Usage: Package Global Data.

Event_Table : Event_Table_Type;
Description: Table of events used by senario_file event.
Package: Event_Table_Obj_Pkg body.
Usage: Package Global Data.

EVENT_TIME : in ATSG_Gen_Pkg.Event_Time_Type;
Description: The time of this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Formal Parameter.

EVENT_TIME : in ATSG_Gen_Pkg.Event_Time_Type;
Description: The time of this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

EVENT_TIME : in ATSG_Gen_Pkg.Event_Time_Type;
Description: The time of this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

EVENT_TIME : in ATSG_Gen_Pkg.Event_Time_Type;
Description: The time of this event.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Formal Parameter.

First : constant INTEGER := 1;
Description: PI [radians] / 2.0
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

FLAT_RANGE : FLOAT;
Description: The AC to target flat range to be used in bearing calculations.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Local Data.

FLAT_RANGE : FLOAT;
Description: The AC to target flat range.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Free_Form_Value :
DBDTBM_FREE_FORM_INCLUDE.DBDTBM_FREE_FORM_REC_TYPE;
Description: Value returned from the free form utility for the value representing the true target track as a free form image in the form of a broken line.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Free_Form_Value :
in DBDTBM_FREE_FORM_INCLUDE.DBDTBM_FREE_FORM_REC_TYPE;

Description: Value returned from the free form utility for the value representing the true target track as a free form image in the form of a broken line.
Package: Free_Form_Obj_Pkg.
Unit: Write_Free_Form_Obj.
Usage: Formal Parameter.

GAC : GAC_TYPE;
Description: The initial lat/lon of the P3 UIV aircraft defines the center of our gaming area (0 X coordinate, 0 Y coordinate). In the NED system North and East are Positive while South and West are Negative. The Sin and Cos of the lat/lon are computed in STORE_POSITION_AND_STEERING_COMMANDS and used by ESTABLISH_INITIAL_LAT_LON and UPDATE_LAT_LON.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

GAC_XYZ : GAC_XY_TYPE;
Description: An array containing the local gaming area x, y, z coordinates. The x, y values are set by this process for the P3 aircraft and by ESTABLISH_INITIAL_LAT_LON for the other target platforms. It is updated by the procedures UPDATE_LINE and UPDATE_ARC. The perturbed x, y positions are stored as the xp, yp items respectively.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

GAC_XYZ : in out GAC_XY_TYPE;
Description: An array containing the local gaming area x, y, z coordinates. The x, y values are set by this process for the P3 aircraft and by ESTABLISH_INITIAL_LAT_LON for the other target platforms. It is updated by the procedures UPDATE_LINE and UPDATE_ARC. The perturbed x, y positions are stored as the xp, yp items respectively.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Formal Parameter.

Ground_Range : SYGLME_MEASURE_TYPE_INCLUDE.SYGLME_NMILES_TYPE;
Description: Ground range A/C to target from PMM to Scenario_Obj for sensor packages (NM).
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Half_Deltime : ATSG_Gen_Pkg.Event_Time_Type;
Description: Half of maximum plot update interval.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Half_Diff : short_float;
Description: The distance from the center point to the extreme side of the plot data.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

Half_speed : constant FLOAT := 0.5;
Description: Multiplied by the speed to get half of the speed.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Local Data.

HEADING : in FLOAT;
Description: Platform heading.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

HEADING : in out FLOAT;
Description: Platform heading.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

Heading : float;
Description: A/C Heading in degrees.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

ID : ATSG_Gen_Pkg.Object_ID_Type
Description: The Object ID of the IRDS object being written.
Package: IRDS_Object_Out_Pkg.
Unit: IRDS_UPD_Header.
Usage: Formal Parameter.

ID : ATSG_Gen_Pkg.Object_ID_Type
Description: The Object ID of the MAD object being written.
Package: MAD_Object_Out_Pkg.
Unit: MAD_UPD_Header.
Usage: Formal Parameter.

ID : ATSG_Gen_Pkg.Object_ID_Type
Description: The Object ID of the Man_Radar object being written.
Package: Man_Radar_Object_Out_Pkg.
Unit: Man_Radar_UPD_Header.
Usage: Formal Parameter.

ID : ATSG_Gen_Pkg.Object_ID_Type;
Description: The Object ID of the TWS_Radar object being written.
Package: TWS_Radar_Object_Out_Pkg.
Unit: TWS_Radar_UPD_Header.
Usage: Formal Parameter.

ID : ATSG_Gen_Pkg.Object_ID_Type;
Description: The Object ID of the Visual object being written.
Package: Visual_Object_Out_Pkg.
Unit: Visual_UPD_Header.
Usage: Formal Parameter.

ILEFT : natural;
Description: The character index for Left as it is being used.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Local Data.

Init_flag : boolean;
Description: Indicator of whether processing is beyond the first call of Write_Plot_File in the load.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Input_Category : Platform_Movement_Model.Input_Category_Type;
Description: Latest Event category for Platform_Movement_Model.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Input_Category_Error : exception;
Description: An exception in PMM data preparation relating to the PMM event category.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

INPUT_RECORD : in INPUT_RECORD_TYPE;
Description: Input records are passed to Store Data by the Input File Object. Store Data then stores the various records in the appropriate data structures for use by the PLATFORM_MOVEMENT_MODELs Object.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Formal Parameter.

INPUT_RECORD : INPUT_RECORD_TYPE;
Description: Input records are passed to Store Data by the Input File Object. Store Data then stores the various records in the appropriate data structures for use by the PLATFORM_MOVEMENT_MODELs Object.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

IOUT : natural;
Description: The character index for Outval as it is being built.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Local Data.

IOUT : natural;
Description: The character index for Outval as it is being built.
Package: Pack_Strings_Pkg.
Unit: pack_strings.
Usage: Local Data.

IRDS_Counts : Update_Count_Type;
Description: Count of IRDS contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

IRDS_Flag : boolean;
Description: Indicates whether IRDS "withs" are needed for SCENARIO.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

IRDS_Object_Counts : Object_Count_Type;
Description: Object ID's of IRDS contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

IRDS_SS : float;
Description: perturbation in ft
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

IRDS_Value :
DBDTEU_TRACK_IRDS_INCLUDE.DBDTEU_TRACK_IRDS_REC_TYPE
Description: The IRDS tactical object to be written to the scenario file.
Package: IRDS_Object_Out_Pkg.
Unit: IRDS_Out.
Usage: Formal Parameter.

IRDS_Value : DBDTEU_Track_IRDS_Include.DBDTEU_Track_IRDS_REC_TYPE;
Description: Value returned from the IRDS utility for the value of the scenario file IRDS object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

IRIGHT : natural;
Description: The character index for Right as it is being used.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Local Data.

IRIGHT : natural;
Description: The character index for Right as it is being used.
Package: Pack_Strings_Pkg.
Unit: pack_strings.
Usage: Local Data.

Itime : Integer;
Description: Dummy interger to print.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Knots_to_ft_per_sec : constant FLOAT := 6076.0 / 3600.0;
Description: [NM]/[HR] * 6076[ft]/1[NM] * 1[Hour] / 3600[sec]
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

Last_AC_Update_Time : ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of last AC object update.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

LAST_ARC_BEGIN_TIME : FLOAT;
Description: Used by Store Positions and Steering Commands to hold the value of the beginning time of an arc segment (sec).
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

LAST_ARC_DIRECTION : DIRECTION_TYPE;
Description: Used by Store Positions and Steering Commands to hold the value of the direction of an arc segment (n/a).
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

LAST_ARC_DURATION : FLOAT;
Description: Used by Store Positions and Steering Commands to hold the value of the duration of an arc segment (sec).
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

LAST_ARC_NO_OF_LOOPS : INTEGER;
Description: Used by Store Positions and Steering Commands to hold the value of the number of loops for an arc segment (n/a).
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

LAST_ARC_RADIUS : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the radius of an arc segment (ft).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

LAST_ARC_THETA : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the direction of an arc segment (radians).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

Last_Input_Category : Platform_Movement_Model.Input_Category_Type;

Description: Latest Event category for Platform_Movement_Model.

Package: Input_File_Obj_Pkg.

Unit: Read_File.

Usage: Local Data.

LAST_LINE_HEADING : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the heading of the previous line segment (radians).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

```
LAST_POSITION_EVENT        : PLATFORM_UPDATE_TYPE :=
        (Aircraft .. ATSG_Gen_Pkg.Max_Target => 0.0) ;
```

Description: Initialized to zero, the last position event is used to determine if the platform position should be updated. If the event time is less than the last position event time, then the position is updated. But if the event time is greater than the last position event time, then the postions remain the same.

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

Last_Time : ATSG_Gen_Pkg.Event_Time_Type;
Description: Previous data time used for checking for decreasing values.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

LAT : out LAT_TYPE;
Description: Latitude.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: GET_GAC.
Usage: Formal Parameter.

LAT : out LAT_TYPE;
Description: Latitude.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Formal Parameter.

Lat : float;
Description: Latest target position for Platform_Movement_Model.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

LAT_GAC : Platform_Movement_Model.LAT_TYPE;
Description: GAC position.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Local Data.

Lat_Rad : in short_float;
Description: Latitude in radians to convert to nm north or equator.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Formal Paramater.

Left : string
Description: The left string to be concatenated pack_strings.
Package: Pack_Strings_Pkg.
Unit: pack_strings.
Usage: Formal Parameter.

Left : string;
Description: The left string to be concatenated by Build_Suffix.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Formal Paramater.

line : in String
Description: A string of text to write to the Scenario file.
Package: Write_Files_Pkg.
Unit: Write_Scenario_File.
Usage: Formal Parameter.

line : in String;
Description: Provides the line of date to be written to the file.
Package: Write_Files_Pkg.
Unit: Write_To_Declare_File.
Usage: Formal Parameter.

line : in String;
Description: provides the line of date to be written to the file.
Package: Write_Files_Pkg.
Unit: Write_to_Command_File.
Usage: Formal Parameter.

Line_Count : in out integer
Description: Previously used to number the schedule lines.
Package: AC_Obj_Pkg.
Unit: Write_AC_Obj.
Usage: Formal Parameter.

Line_Count : in out integer
Description: Previously used to number the schedule lines.
Package: AC_State_Pkg.
Unit: Write_AC_State.
Usage: Formal Parameter.

Line_Count : in out integer
Description: The statement line number for the previous line in the Command_File.
Package: Init_SF_Pkg.
Unit: Write_CR_Init.
Usage: Formal Parameter.

Line_Count : in out integer
Description: The statement line number for the previous line in the Command_File.
Package: Init_SF_Pkg.
Unit: Write_General_Types .
Usage: Formal Parameter.

Line_Count : in out integer
Description: The statement line number for the previous line in the Command_File.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Formal Parameter.

Line_Count : in out integer
Description: Used to number the schedule lines.
Package: Mark_Time_Object_Out_Pkg.
Unit: Mark_Time_OUT.
Usage: Formal Parameter.

Line_Count : in out integer
Description: Used to number the schedule lines.
Package: MOT_Object_Out_Pkg.
Unit: MOT_OUT.
Usage: Formal Paramater.

Line_Count : in out integer
Description: Used to number the schedule lines.
Package: Schedule_Pkg.
Unit: Schedule.
Usage: Formal Parameter.

Line_Count : in out integer;
Description: Used to number the schedule lines. Incremented by 5 immediately before each line is added.
Package: Free_Form_Obj_Pkg.
Unit: Write_Free_Form_Obj.
Usage: Formal Parameter.

Line_Count : integer;
Description: The statement line number for the Command_File. Incremented by 5 immediately before each line is added.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

LINESIZE : constant := 79;
Description: Size of line.
Package: Write_Files_Pkg spec.
Usage: ATSG Global Data.

List_Set_Type : ESM_Table_Obj_Pkg.Plat_List_Set_T_type;
Description: Describes the contents of the platform name list. Both the UNIVERSAL and EMPTY lists are arrays of null strings but they are handled differently in classification scoring and fusion.
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Local Data.

Local_Flt : float;
Description: An float read from a file.
Package: Input_File_Obj_Pkg body.
Usage: Package Global Data.

Local_Int : integer;
Description: An integer read from a file.
Package: Input_File_Obj_Pkg body.
Usage: Package Global Data.

LON : out LON_TYPE;
Description: Longitude.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: GET_GAC.
Usage: Formal Parameter.

LON : out LON_TYPE;
Description: Longitude.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Formal Parameter.

Lon : float;
Description: Latest target position for Platform_Movement_Model.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

LON_GAC : Platform_Movement_Model.LON_TYPE;
Description: GAC position.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Local Data.

Lon_Rad : in short_float;
Description: Longitude in radians to convert to nm east of prime meridian.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Formal Paramater.

MAD_Counts : Update_Count_Type;
Description: Count of MAD contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

MAD_Flag : boolean;
Description: Indicates whether MAD "withs" are needed for SCENARIO.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

MAD_Object_Counts : Object_Count_Type;
Description: Object ID's of MAD contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

MAD_SS : float;
Description: Perturbation in feet.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

MAD_Track : DBDTEV_Track_MAD_Include.DBDTEV_Track_MAD_Rec_Type;
Description: Value returned from the MAD utility for the value of the scenario file MAD object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Mad_Value : DBDTEV_TRACK_MAD_INCLUDE.DBDTEV_TRACK_MAD_REC_TYPE
Description: The MAD tactical object to be written to the scenario file.
Package: MAD_Object_Out_Pkg.
Unit: MAD_Out.
Usage: Formal Parameter.

Man_Radar_Counts : Update_Count_Type;
Description: Count of Radar contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Man_Radar_Flag : boolean;
Description: Indicates whether Man_Radar "withs" needed for SCENARIO.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Man_Radar_Object_Counts : Object_Count_Type;
Description: Object ID's of Manual Radar contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Man_Radar_Value : DBDTEX_Track_Manual_Radar_Include.
DBDTEX_Track_Manual_Radar_Rec_Type;
Description: Value returned from the Radar utility for the value of the scenario file Radar object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Man_Radar_Value :
DBDTEX_TRACK_Manual_Radar_INCLUDE.DBDTEV_TRACK_MAD_REC_TYP
E
Description: The Man_Radar tactical object to be written to the scenario file.
Package: Man_Radar_Object_Out_Pkg . **Unit**: Man_Radar_Out.
Usage: Formal Parameter.

MAN_RDR_SS : float;
Description: Perturbation in feet.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

MANEUVER_EX_TIME : ATSG_Gen_Pkg.Event_Time_Type;
Description: The length of time (sec) for this update.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Local Data.

Max_DOIT_Events : constant := 1000;
Description: Max number of DOIT events.
Package: D_OIT_Obj_Pkg body.
Usage: Package Global Data.

Max_DSIT_Events : constant := 1000;
Description: Max number of DSIT events.
Package: D_SIT_Obj_Pkg body.
Usage: Package Global Data.

Max_DSRT_Events : constant := 1000;
Description: Max number of DSRT events.
Package: D_SRT_Obj_Pkg body.
Usage: Package Global Data.

Max_Duration : constant FLOAT := 864000.0;
Description: The maximum length of time (sec) for this mission.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_Events : Event_Count_Type;
Description: The maximum number of events for this game.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data.

Max_LAT : constant FLOAT := PI/2.0; -- radians
Description: Maximum Latitude.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_LON : constant FLOAT := PI; -- radians
Description: Maximum Longitude.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_Loop : constant FLOAT := 100.0;
Description: Maximum number of loops per maneuver.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_Seg_ID : constant INTEGER := 50;
Description: Maximum number of maneuver segments.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_Target : constant := 6;
Description: The maximum number of targets which can be simulated by ATSG.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data.

Max_target : constant INTEGER := 6;
Description: Maximum number of targets in ATSG.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Max_Time : Event_Time_Type;
Description: The maximum game time in seconds.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data.

MAXEMITTERS : constant := 20;
Description: The maximum number of emitters for a target.
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data.

MAXEVENTS : constant := 10000;
Description: Thge maximum number of events in the event table.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data.

MAXp1 : constant := MAXsize+1;
Description: The maximum size of the output string used as input to Quote_Terminate.
Package: Pack_Strings_Pkg spec.
Usage: ATSG Global Data.

MAXsize : constant := 6;
Description: The maximum size of the input string used as input to Quote_Terminate.
Package: Pack_Strings_Pkg spec.
Usage: ATSG Global Data.

MEASUREMENT_DATA : in CRCRGLCR_CORRELATION_TYPE_INCLUDE.
CRCRGLCR_FIX_RECORD_TYPE
Description: The MEASUREMENT_DATA portion of the tactical object to be written to the scenario file.
Package: Measurement_Data_Out_Pkg.
Unit: Write_Measurement_Data.
Usage: Formal Parameter.

Min_LAT : constant FLOAT := -PI/2.0; -- radians
Description: Minimum Latitude.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Min_LON : constant FLOAT := -PI; -- radians
Description: Minimum Longitude.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data.

Model_Kind : Model_Kind_Type;
Description: The type of model being run.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data.

More_Platform_Data : boolean;
Description: Indicates platform data is present.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

MOT_Value :
DBDTDE_MARK_ON_TOP_INCLUDE.DBDTDE_MARK_ON_TOP_REC_TYPE;
Description: Initial MOT data.
Package: MOT_Object_Out_Pkg.
Unit: MOT_OUT.
Usage: Formal Paramater.

MOT_Value :
DBDTDE_MARK_ON_TOP_INCLUDE.DBDTDE_MARK_ON_TOP_REC_TYPE;
Description: Object to be written to the scenario file to represent a MOT object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

MS_Delay : in integer;
Description: Time to wait in mileseconds after SCHEDULE.
Package: Schedule_Pkg.
Unit: Schedule.
Usage: Formal Parameter.

Name : in string;
Description: string to place on the scenario_file.
Package: Schedule_Pkg.
Unit: Schedule.
Usage: Formal Parameter.

NEXT_ARC_BEGIN_HEADING : TARGET_HDG_ARRAY_TYPE;
Description: Set by preceding line segment and used by the arc segment.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

NEXT_BEGIN_SEG : SEGMENT_ID_TYPE;
Description: Set to the segment for the beginning of the next event update.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Local Data.

NEXT_SPEED : FLOAT;
Description: Used by Store Positions and Steering Commands to hold the value of the beginning speed of a segment (ft/sec).
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

NM_to_ft : constant FLOAT := 6076.0 / 1.0;
Description: 6076 [ft] / 1 [NM].
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

NMprad : constant := 10800.0/3.1415927;
Description: The number of Nautical miles in a radian of latitude.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

NO_OF_LOOPS : integer;
Description: Number of 360 degree turns after reaching destination heading.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Not_Terminated : boolean;
Description: An indicator of whether the double quote was placed on other than the last (right most) character of the output string.
Package: Pack_Strings_Pkg.
Unit: Quote_Terminate.
Usage: Local Data.

Number : in integer;
Description: Target number to put in 3rd column (unless 0 or Time < 0).
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Formal Paramater.

One_Char : character;
Description: A single character buffer for copying file to file.
Package: Write_Files_Pkg.
Unit: Copy_Command_File.
Usage: Local Data.

One_Char : character;
Description: A single character buffer for copying file to file.
Package: Write_Files_Pkg.
Unit: Copy_Declare_File.
Usage: Local Data.

Orgx : short_float;
Description: The origin or the position of the initial point to be plotted.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Orgy : short_float;
Description: The origin or the position of the initial point to be plotted.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Outval : string (1 .. Left'last + Right'last-1);
Description: The string value to build and return by pack_strings.
Package: Pack_Strings_Pkg.
Unit: pack_strings.
Usage: Local Data.

Outval : String_7_Type;
Description: The string value to build and return by Quote_Terminate.
Package: Pack_Strings_Pkg.
Unit: Quote_Terminate.
Usage: Local Data.

Outval : String_Six_Type;
Description: The string value to build and return by Build_Suffix.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Local Data.

Package_Name : in String;
Description: The package name for which with and use clauses will be written to the scenario file.
Package: Write_Files_Pkg.
Unit: Write_Scenario_File_WU.
Usage: Formal Parameter.

PI_over_2 : constant FLOAT := PI / 2.0;
Description: PI [radians] / 2.0.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

PLATFORM : in ATSG_Gen_Pkg.Target_Num_Type;
Description: Platform number.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

PLATFORM : in ATSG_Gen_Pkg.Target_Num_Type;
Description: Platform number.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

Platform : ESM_Table_Obj_Pkg.Platform_Type;
Description: A NATO class of a target.
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Local Data.

Platform_Data : text_io.file_type;
Description: LFN for the platform_data.dat.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

PLATFORM_LAST_UPDATE : PLATFORM_UPDATE_TYPE :=
(Aircraft .. ATSG_Gen_Pkg.Max_Target => 0.0) ;
Description: Initialized to zero, the last update time is used to determine if the platform position should be updated for the current event time.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

Plot_file_exception : exception;
Description: An unexpected exception in Plot_File_Pkg.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Plot_LFN : Text_io.file_type;
Description: Plot file local file name.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Plot_Time : ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of last AC object plot update.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

PMM_POSITION_INFO : PLATFORM_MOVEMENT_MODEL.POSITION_INFO_TYPE;
Description: Position data from PMM to Scenario_Obj for sensor packages.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

POIT_Event : out P_OIT_Obj_Pkg.POIT_Event_Type
Description: Data to describe a POIT event.
Package: Input_File_Obj_Pkg.
Unit: Read_POIT_Event.
Usage: Formal Parameter.

POIT_Event : P_OIT_Obj_Pkg.POIT_Event_Type;
Description: Probabilistic Operator Input events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

POS_INFO : out POSITION_INFO_TYPE;
Description: The Position info record is returned to the Scenario File Object with the data needed to write the true position vectors and the data needed to send to the sensor utilities to create a target detection.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Formal Parameter.

POSITION_INFO : POSITION_INFO_TYPE;
Description: The Position info record is returned to the Scenario File Object with the data needed to write the true position vectors and the data needed to send to the sensor utilities to create a target detection.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

Premature_SRT : boolean;
Description: Indicates whether DSRT data is too late to use.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

Prev_Time : ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of previous event.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

Prev_Time : in ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of previous event extracted from any source.
Package: D_OIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Prev_Time : in ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of previous event extracted from any source.
Package: D_SIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

Prev_Time : in ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of previous event extracted from any source.
Package: D_SRT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Formal Parameter.

PSIT_Event : out P_SIT_Obj_Pkg.PSIT_Event_Type
Description: Data to describe a PSIT event.
Package: Input_File_Obj_Pkg.
Unit: Read_PSIT_Event.
Usage: Formal Parameter.

PSIT_Event : P_SIT_Obj_Pkg.PSIT_Event_Type;
Description: Probabilistic Sensor Input events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

PSRT_Event : out P_SRT_Obj_Pkg.PSRT_Event_Type
Description: Data to describe a PSRT event.
Package: Input_File_Obj_Pkg.
Unit: Read_PSRT_Event.
Usage: Formal Parameter.

PSRT_Event : P_SRT_Obj_Pkg.PSRT_Event_Type;
Description: Probabilistic Situation Response events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Psuffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the delete declaration value.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Quote : character;
Description: Used to put double quotes on scenario file.
Package: Pack_Strings_Pkg.
Unit: Quote_Terminate.
Usage: Local Data.

Quote : String (1..1);
Description: Single character string holding the double quote.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Quote : string (1 .. 1);
Description: Used to put double quotes on scenario file.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: A double quote character used for building strings.
Package: AC_State_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Local Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: IRDS_Object_Out_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: MAD_Object_Out_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: Man_Radar_Object_Out_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: TWS_Radar_Object_Out_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Description: Single character string holding the double quote.
Package: Visual_Object_Out_Pkg body.
Usage: Package Global Data.

Quote : string (1..1);
Package: AC_Obj_Pkg body.
Usage: Package Global Data.

R_GEO_ERROR : in BOOLEAN;
Description: The geometric error flag.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Formal Parameter.

Rad_Per_Deg : constant := 0.01745329277778;
Description: Radians per degree conversion factor.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

RADIUS : in FLOAT;
Description: The radius of the arc.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

Radius : float;
Description: The turn radius for a PMM turn in nm.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

RADIUS_TO_TURN_ORG : FLOAT;
Description: The Radius to the turn origin from the GAC.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Local Data.

Read_File local
Steering_Command : Platform_Movement_Model.INPUT_RECORD_TYPE;
Description: The steering record to store in the PMM object.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Response : Operator_Pkg.Situation_Response_Type;
Description: The operator response selection being read and processed.
Package: Object_Manager_Pkg.
Unit: Get_Operator_Response_Characteristics.
Usage: Formal Paramater.

Response_Object_ID_Array : Response_Object_ID_Array_Type := (others => 0);
Description: An indication of how many additional objects are generated by the responses.
Package: Object_Manager_Pkg spec.
Usage: ATSG Global Data.

Right : string
Description: The right string to be concatenated pack_strings.
Package: Pack_Strings_Pkg.
Unit: pack_strings.
Usage: Formal Parameter.

right : string;
Description: The right string to be concatenated by Build_Suffix.
Package: Pack_Strings_Pkg.
Unit: Build_Suffix.
Usage: Formal Paramater.

Scale_Strx : string (1 .. 20);
Description: String used to hold the ASCII representations of plot position X and Y components.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

Scale_Stry : string (1 .. 20);
Description: String used to hold the ASCII representations of plot position X and Y components.
Package: Plot_File_Pkg.
Unit: Close_Plot_File.
Usage: Local Data.

Scenario_file_exception : exception;
Description: An error in writing the scenario file.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

Scenario_LFN : Text_io.file_type;
Description: Scenario file local file name.
Package: Write_Files_Pkg body.
Usage: Package Global Data.

SDR : FLOAT;
Description: Standard Deviation of the Range.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Searching : boolean;
Description: An indicator of whether the search for the appropriate event to return can continue.
Package: D_OIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Local Data.

Searching : boolean;
Description: An indicator of whether the search for the appropriate event to return can continue.
Package: D_SRT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Local Data.

Searching : boolean;
Description: Count or index of DSIT event.
Package: D_SIT_Obj_Pkg.
Unit: Get_Next_Event.
Usage: Local Data.

Segment_ID : Platform_Movement_Model.Segment_ID_Type;
Description: Latest sequence number for Platform_Movement_Model.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

SENSOR_SENSITIVITY : array(ATSG_gen_Pkg.EVENT_KIND_TYPE) of FLOAT;
Description: Initialized by Store Positions and Steering Commands and used as the maximum permissible deviation in feet for the perturbed target position as detected by a sensor.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

SI_OI_Indicator : SI_OI_Indicator_Type;
Description: Indicates whether previous event selected was SI vs OI.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

Slant_Range : SYGLME_MEASURE_TYPE_INCLUDE.SYGLME_NMILES_TYPE;
Description: Slant range A/C to target from PMM to Scenario_Obj for sensor packages (NM).
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

SPEED : in FLOAT;
Description: Speed in ft/ sec.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

SPEED : in out FLOAT;
Description: Speed in ft/ sec.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

Speed : float range 0.0 .. 99999.0;
Description: The speed of a platform at the start of the initial segment in knots.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

SS_VALUE_DEFAULT : array(ATSG_gen_Pkg.EVENT_KIND_TYPE) of BOOLEAN;
Description: Initialized by Store Positions and Steering Commands to TRUE if the input sensor sensitivity type is equal to -1.0 and is initialized to FALSE if the input sensor sensitivity is greater than -1.0.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

STEERING_CMD : array (Aircraft .. ATSG_Gen_Pkg.Max_Target, 1 .. Max_Seg_ID)
of STEERING_CMD_TYPE;
Description: Steering Commands are derived from the input records and stored in an array of Steering Command Type Records to be used by PLATFORM_MOVEMENT_MODEL modules to update the positions of the platforms.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

Steering_Cmd : out Platform_Movement_Model.INPUT_RECORD_TYPE;
Description: Data to describe a platform event.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Formal Parameter.

Steering_Cmd : Platform_Movement_Model.INPUT_RECORD_TYPE;
Description: Platform Movement Model events to store.
Package: Input_File_Obj_Pkg.
Unit: Read_File.
Usage: Local Data.

Str2 : string (1 .. 20);
Description: A string used to store ASCII representations of real values.
Package: AC_State_Pkg body.
Usage: Package Global Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Local Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: IRDS_Object_Out_Pkg.
Unit: IRDS_Out.
Usage: Local Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Mark_Time_Object_Out_Pkg body.
Usage: Package Global Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Measurement_Data_Out_Pkg.
Unit: Write_Measurement_Data.
Usage: Local Data.

Str2 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: AC_Obj_Pkg body.
Usage: Package Global Data.

Str20 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: MOT_Object_Out_Pkg.
Unit: MOT_OUT.
Usage: Local Data.

Str20 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Track_Data_Out_Pkg.
Unit: Write_Track_Data.
Usage: Local Data.

Str20 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: TWS_Radar_Object_Out_Pkg.
Unit: TWS_Radar_Out.
Usage: Local Data.

Str20 : string (1 .. 20);
Description: String used to hold the ASCII representation of real values.
Package: Visual_Object_Out_Pkg.
Unit: Visual_Out.
Usage: Local Data.

str20   : string (1 .. 20);
Description: String to hold formatted data.
Package: MAD_Object_Out_Pkg.
Unit: MAD_Out.
Usage: Local Data.

str20   : string (1 .. 20);
Description: String to hold formatted data.
Package: Man_Radar_Object_Out_Pkg.
Unit: MAD_Out.
Usage: Local Data.

Strin : in   String_Six_Type;
Description: The string on which to append a double quote.
Package: Pack_Strings_Pkg.
Unit: Quote_Terminate.
Usage: Formal Paramater.

Strlat : string (1 .. 20);
Description: String used to hold the ASCII representation of display position latitude for Targ_LFN.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Strlon : string (1 .. 20);
Description: String used to hold the ASCII representation of display position longitude for Targ_LFN.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Strx : string (1 .. 20);
Description: String used to hold the ASCII representations of plot position X and Y components.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Stry : string (1 .. 20);
Description: String used to hold the ASCII representations of plot position X and Y components.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: AC_Obj_Pkg body.
Usage: Package Global Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: AC_State_Pkg body.
Usage: Package Global Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: Mark_Time_Object_Out_Pkg body.
Usage: Package Global Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: MOT_Object_Out_Pkg.
Unit: MOT_OUT.
Usage: Local Data.

Suffix : Pack_Strings_Pkg.String_Six_Type;
Description: Temporary buffer to hold the suffix to the declaration.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Same as Suffix except double quote is added.
Package: Mark_Time_Object_Out_Pkg body.
Usage: Package Global Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Same as Suffix except double quote is added.
Package: MOT_Object_Out_Pkg.
Unit: MOT_OUT.
Usage: Local Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Temporary buffer to hold the suffix to the declaration including a double quote.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Temporary buffer to hold the suffix to the declaration terminated by a double quote.
Package: AC_State_Pkg body.
Usage: Package Global Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Temporary buffer to hold the suffix to the declaration. This buffer is terminated by a quote and blanks after the last non-blank char.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Suffixq : Pack_Strings_Pkg.String_7_Type;
Description: Temporary buffer to hold the suffix to the declaration. This buffer is terminated by a quote and blanks after the last non-blank char.
Package: AC_Obj_Pkg body.
Usage: Package Global Data.

TACTICAL_PLANE_REC : UTGETO_TACTICAL_ORIGIN_INCLUDE.

UTGETO_TACTICAL_ORIGIN_REC_TYPE;
Description: Result of call to set tactical plane.
Package: Init_SF_Pkg.
Unit: Write_Tac_Init.
Usage: Local Data.

Targ_LFN : Text_io.file_type;
Description: Target file local file name.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

TARGET : in ATSG_Gen_Pkg.Target_Num_Type;
Description: Target number.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Formal Parameter.

TARGET : in ATSG_Gen_Pkg.Target_Num_Type;
Description: Target number.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Formal Parameter.

TARGET : in ATSG_Gen_Pkg.Target_Num_Type;
Description: Target number.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Formal Parameter.

Target : ATSG_Gen_Pkg.Target_Index_Type;
Description: Target number being declared.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Target_Category : ESM_Table_Obj_Pkg.Target_Category_Type;
Description: A description of the category of target (Subsurface, Surface, Unknown, Air).
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Local Data.

Target_Count : ATSG_Gen_Pkg.Target_Num_Type;
Description: The count of the targets with emitter data being stored.
Package: ESM_Table_Obj_Pkg body.
Usage: Package Global Data.

Target_Count_Array : Target_Count_Array_Type := (others => 0);
Description: Count of contacts for all targets for all sensors. The first contact for each target is numbered 1.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data.

Target_ID : in ATSG_Gen_Pkg.TARGET_Index_TYPE;
Description: A target identification number (0 is the A/C).
Package: ESM_Table_Obj_Pkg.
Unit: Get_Data.
Usage: Formal Parameter.

Target_Index : ATSG_Gen_Pkg.Target_Index_Type;
Description: Index to the ESM data array.
Package: ESM_Table_Obj_Pkg.
Unit: Get_Data.
Usage: Local Data.

TARGET_LAT : array(Aircraft .. ATSG_Gen_Pkg.Max_Target) of LAT_TYPE;
Description: Establishes the initial latitude of each target. From the latitude the initial local gaming area coordinate system x-y coordinates can are calculated. Used by UPDATE_LAT_LON procedure and ESTABLISH_INITIAL_LAT_LON.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

TARGET_LON : array(Aircraft .. ATSG_Gen_Pkg.Max_Target) of LON_TYPE;
Description: Establishes the initial longitude of each target. From the longitude the initial local gaming area coordinate system x-y coordinates can are calculated. Used by UPDATE_LAT_LON procedure and ESTABLISH_INITIAL_LAT_LON.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

Target_Number : ATSG_Gen_Pkg.Target_Num_Type;
Description: Latest target number for procedure used. Remember between calls to Read_Platform_Event so it does not have to be re-read.
Package: Input_File_Obj_Pkg body.
Usage: Package Global Data.

Target_Table : Target_Table_Type;
Description: The table of target types, classes, and number of emitters.
Package: ESM_Table_Obj_Pkg body.
Usage: Package Global Data.

TGT_LAT : in FLOAT;
Description: Establishes the initial latitude of each target. From the latitude the initial local gaming area coordinate system x-y coordinates can are calculated. Used by UPDATE_ LAT_LON procedure and ESTABLISH_INITIAL_LAT_LON.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Formal Parameter.

TGT_LON : in FLOAT;
Description: Establishes the initial longitude of each target. From the longitude the initial local gaming area coordinate system x-y coordinates can are calculated. Used by UPDATE_LAT_LON procedure and ESTABLISH_INITIAL_LAT_LON.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Formal Parameter.

Tgt_Position_Info :
SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
Description: Target pos data fm PMM to Scenario_Obj for sensor packages.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

THETA in FLOAT;
Description: The angle.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

THIS_AC_DC_RATE : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the ac/dc rate of a line segment (ft/sec**2).

Package: PLATFORM_MOVEMENT_MODEL.

Unit: STORE_POSITION_AND_STEERING_COMMANDS.

Usage: Local Data.

THIS_BEGIN_TIME : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the beginning time of a line segment (sec).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

THIS_DURATION : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the duration of a line segment (sec).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

THIS_HEADING : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the heading of a segment (end for arc) (radians).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

THIS_SPEED : FLOAT;

Description: Used by Store Positions and Steering Commands to hold the value of the beginning speed of a line segment (ft/sec).

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data.

Threat_Class : ESM_Table_Obj_Pkg.Threat_Class_Type;

Description: A description of the class of target (Friendly, Hostile, Unknown).

Package: Input_File_Obj_Pkg.

Unit: Read_Emitter_Data.

Usage: Local Data.

Time : in ATSG_Gen_Pkg.Event_Time_Type;
Description: Time to reset system clock.
Package: Mark_Time_Object_Out_Pkg.
Unit: Mark_Time_OUT.
Usage: Formal Parameter.

Time : in short_float;
Description: Time to put in 3rd column (unless negative).
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Formal Paramater.

Time_Now : ATSG_Gen_Pkg.Event_Time_Type;
Description: Time of latest DSIT or DOIT event.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data.

Track_Data : in CRCRGLCR_CORRELATION_TYPE_INCLUDE.
CRCRGLCR_TRACK_RECORD_TYPE
Description:The component of a tactical object associated with the track.
Package: Track_Data_Out_Pkg.
Unit: Write_Track_Data.
Usage: Formal Parameter.

True_Tgt_Pos_Info :
SYGLGE_GEOGRAPHIC_TYPE_INCLUDE.SYGLGE_POSITION_TYPE;
Description: Target true position data fm PMM for free form package.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Two_NM : constant FLOAT := 6076.0 * 2.0;
Description: Two NM's multiplied by a random number between 1 and 2 to determine the geometric error position.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Local Data.

Two_PI : constant FLOAT := PI * 2.0;
Description: PI [radians] * 2.0.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data.

Two_PI : constant FLOAT := PI * 2.0;
Description: PI [radians] * 2.0.
Package: PLATFORM_MOVEMENT_MODEL
Unit: STORE_POSITION_AND_STEERING_COMMANDS.
Usage: Local Data.

TWS_Radar_Counts : Update_Count_Type;
Description: Count of TWS Radar contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

TWS_Radar_Flag : boolean;
Description: Indicates whether Man_Radar "withs" needed for SCENARIO.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

TWS_Radar_Object_Counts : Object_Count_Type;
Description: Object ID's of TWS Radar contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

TWS_Radar_Value : DBDTEY_TRACK_TWS_Radar_INCLUDE.
DBDTEY_TRACK_TWS_Radar_REC_TYPE

Description: Object to be written to the scenario file to represent a TWS_Radar object.
Package: TWS_Radar_Object_Out_Pkg.
Unit: TWS_Radar_Out.
Usage: Formal Parameter.

TWS_Radar_Value :
DBDTEY_Track_TWS_Radar_Include.DBDTEY_Track_TWS_Radar_REC_TYPE;

Description: Value returned from the TWS_RADAR utility for the value of the scenario file TWS_RADAR object.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

TWS_RDR_SS : float;

Description: Perturbation in feet.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Update_Object_ID : ATSG_Gen_Pkg.Object_ID_Type;

Description: Count of contact ADD objects encountered so far beginning at 1 for the A/C and incremented for each new ADD (other than A/C).
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Update_Object_ID : in ATSG_Gen_Pkg.Object_ID_Type;

Description: Count of contact ADD objects encountered so far beginning at 1 for the A/C and incremented for each new ADD.
Package: Free_Form_Obj_Pkg.
Unit: Write_Free_Form_Obj.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description : True if the Emitter_Event data is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_Emitter_Data.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the DOIT_Event data below is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_DOIT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the DSIT_Event data below is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_DSIT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the DSRT_Event data below is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_DSRT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the POIT_Event data is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_POIT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the PSIT_Event data is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_PSIT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the PSRT_Event data is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_PSRT_Event.
Usage: Formal Parameter.

Valid_Data : out boolean;
Description: True if the Steering_Cmd data is valid.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Formal Parameter.

Visual_Counts : Update_Count_Type;
Description: Count of Visual contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Visual_Flag : boolean;
Description: Indicates whether Visual "withs" needed for SCENARIO.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

Visual_Object_Counts : Object_Count_Type;
Description: Object ID's of Visual contacts for all targets.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

VISUAL_SS : float;
Description: Perturbation in feet.
Package: Input_File_Obj_Pkg.
Unit: Read_Platform_Data.
Usage: Local Data.

Visual_Value : DBDTEZ_TRACK_VISUAL_INCLUDE.
DBDTEZ_TRACK_VISUAL_REC_TYPE;

Description: Value returned from the VISUAL utility for the value of the scenario file VISUAL object.

Package: Scenario_File_Obj_Pkg body.

Usage: Package Global Data.

Visual_Value : in DBDTEZ_TRACK_VISUAL_INCLUDE.
DBDTEZ_TRACK_VISUAL_REC_TYPE

Description: Object to be written to the scenario file to represent a Visual contact object.

Package: Visual_Object_Out_Pkg.

Unit: Visual_Out.

Usage: Formal Parameter.

X_ORG : FLOAT;

Description: The x coordinate of the turn origin.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: UPDATE_ARC.

Usage: Local Data.

X_POS : in FLOAT;

Description: The x coordinate of the platform.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: UPDATE_LAT_LON.

Usage: Formal Parameter.

X_POS : in out FLOAT;

Description: The x coordinate of the platform.

Package: PLATFORM_MOVEMENT_MODEL.

Unit: UPDATE_ARC.

Usage: Formal Parameter.

X_POS : in out FLOAT;
Description: The x coordinate of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

X_POS : out FLOAT;
Description: The x coordinate of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Formal Parameter.

X_POS_Aircraft : in FLOAT;
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

X_POS_PLATFORM : in FLOAT;
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

X_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

X_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Local Data.

X_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Local Data.

X_Y_PROJ : FLOAT;
Description: The projection into the x-y plane.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

Xlo : short_float;
Description: Limits on plot values.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

XPOS : FLOAT;
Description: X position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

XX : short_float;
Description: Scaled coordinate to be plotted.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Y_ORG : FLOAT;
Description: The y coordinate of the turn origin.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Local Data.

Y_POS : in FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Formal Parameter.

Y_POS : in out FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_ARC.
Usage: Formal Parameter.

Y_POS : in out FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LINE.
Usage: Formal Parameter.

Y_POS : out FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: ESTABLISH_INITIAL_LAT_LON.
Usage: Formal Parameter.

Y_POS_Aircraft : in FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

Y_POS_PLATFORM : in FLOAT;
Description:Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

Y_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Y_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: MOVE_PLATFORM.
Usage: Local Data.

Y_PRN : FLOAT;
Description: PRN between 0.0 and 1.0.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_PLATFORM_POSITIONS.
Usage: Local Data.

Yards_to_Feet : constant FLOAT := 3.0;
Description: Three Feet per Yard. [Yard] * [3 Ft/Yard] = Ft.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: COMPUTE_DEFAULT_OFFSET.
Usage: Local Data.

Yhi : short_float;
Description: Limits on plot values.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

Ylo : short_float;
Description: Limits on plot values.
Package: Plot_File_Pkg body.
Usage: Package Global Data.

YPOS : FLOAT;
Description: Y position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_LAT_LON.
Usage: Local Data.

YY : short_float;
Description: Scaled coordinate to be plotted.
Package: Plot_File_Pkg.
Unit: Write_Plot_File.
Usage: Local Data.

Z_POS_Aircraft : in FLOAT;
Description: Z position of the aircraft.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

Z_POS_PLATFORM : in FLOAT;
Description: Z position of the platform.
Package: PLATFORM_MOVEMENT_MODEL.
Unit: UPDATE_RANGE_BEARING.
Usage: Formal Parameter.

Zero_Flag : boolean;
Description: Indicates blanking out NATO array.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data.

AUTOMATIC TEST SCENARIO GENERATOR
(ATSG)
DESIGN DESCRIPTION

APPENDIX B: TYPE DEFINITIONS

12 June 1992

```
type Action_Object_ID_Array_Type is array (Operator_Pkg.Action_Type)
                                                  of Number_Of_Objects_Type;
```

Description: An indication of how many additional objects are generated by the actions.
Package: Object_Manager_Pkg spec.
Usage: ATSG Global Data Type.

```
type Action_Type is (Potential_Lost_Contact, Lost_Contact, Initialize_Track,
     Update_Track, Collocation,  -- Notices require no oper response
     MOT_Contact, Enter_Contact, TACCO_Corr_N_C, -- Oper initiated switch hits
     Split_Track, Change_Ctc_Num, Corr_Mode, Manual, Auto, Req_Corr, Trk_Hst,
     Assign_Class, Posit_Uncty, Xtend_Trk_Vec, Fix_Desig, Proj_Posit, Comp_Int,
     Gen_Trk, Assign_NATO_Name, Assign_DI, Coloc_Mod, Trk_Sel, No_Action);
```

Description: A description of the unilateral OIT actions available to the operator wrt sensor correlation/collocation. (Ref CDRL B006 Delivery order #0012, Contract N62269-90-D-0100)
Package: Operator_Pkg spec.
Usage: ATSG Global Data Type.

```
type Attribute_Type is (Fused_Track, MAD_Contact); -- , TBD??
```

Description: The index of the objects which can be acted upon by the operator unilaterally wrt correlation/collocation.
Package: Operator_Pkg spec.
Usage: ATSG Global Data Type.

```
type COMMAND_PTR_TYPE  is
  record
    START_SEGMENT   : SEG_PTR(Aircraft .. ATSG_Gen_Pkg.Max_Target);
    END_SEGMENT     : SEG_PTR(Aircraft .. ATSG_Gen_Pkg.Max_Target);
  end record;
```

Description: Used to define pointers to the proper line/arc segment for each platform.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type COMMAND_TYPE is (LINE, ARC);
```

Description: Used to define line/arc segment for each segment.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type DIRECTION_TYPE is (LEFT, RIGHT);
```

Description: Used to define direction of the turn.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

```
type DOIT_Event_Type is
  record
    Time          :  ATSG_Gen_Pkg.Event_Time_Type;
    Action        : Operator_Pkg.Action_Type;
    Comment       : Operator_Pkg.Comment_Type;
    --Attribute   : Operator_Pkg.Attribute_Type;
  end record;
```

Description: A record of DOIT operator action data.
Package: D_OIT_Obj_Pkg spec.
Usage: ATSG Global Data Type.

```
type DSIT_Event_Type is
  record
    Target_Number      : ATSG_Gen_Pkg.TARGET_NUM_TYPE; -- 0 means A/C
    Time               : ATSG_Gen_Pkg.Event_Time_Type; -- seconds
    Sensor_Category    :  ATSG_Gen_Pkg.Event_Kind_Type;
    Emitter_Number     :  ESM_Table_Obj_Pkg.Emitter_Number_Type;
    P_Det_Flag         : boolean;  -- Detection flag
    Q_Class_Err_Flg    : boolean;  -- Class error flag
    R_OOR_Flag         : boolean;  -- Out_of_range flag
  end record;
```

Description: A record of DSIT sensor event data for a target/sensor
Package: D_SIT_Obj_Pkg spec.
Usage: ATSG Global Data Type.

```
type DSRT_Event_Type is
  record
    Time          : ATSG_Gen_Pkg.Event_Time_Type;
    Action        : Operator_Pkg.Situation_Response_Type;
    Comment       : Operator_Pkg.Comment_Type;
  end record;
```

Description: A record of input DSRT data used to build the Event table.

Package: D_SRT_Obj_Pkg spec.

Usage: ATSG Global Data Type.

```
type Emitter_Count_Type is range 0 .. MAXEMITTERS;
```

Description: The number of emitters for a target.

Package: ESM_Table_Obj_Pkg spec.

Usage: ATSG Global Data Type.

```
type Emitter_Event_Type is
  record
    Target_Number       : ATSG_Gen_Pkg.TARGET_NUM_TYPE; -- 0 means A/C
    Number_Emitters     : Emitter_Count_Type; -- for the target
    Threat_Class        : Threat_Class_Type;
    Target_Category     : Target_Category_Type;
    Platform            : Platform_Array_Type;
    Plat_List_Set_Type  : Plat_List_Set_T_type;
  end record;
```

Description: A record of emitter data for a target

Package: ESM_Table_Obj_Pkg spec.

Usage: ATSG Global Data Type.

```
type Event_Count_Type is range 0 ..  Max_DOIT_Events;
```

Description: Count or index of DOIT events.

Package: D_OIT_Obj_Pkg body.

Usage: Package Global Data Type.

```
type Event_Count_Type is range 0 ..  Max_DSIT_Events;
```

Description: Count or index of DSIT event.

Package: D_SIT_Obj_Pkg body.

Usage: Package Global Data Type.

```
type Event_Count_Type is range 0 ..  Max_DSRT_Events;
```

Description: An index and counter for events to be processed into the DSRT table.

Package: D_SRT_Obj_Pkg body.

Usage: Package Global Data Type.

```
type  Event_Count_Type is range 0 .. MAXEVENTS;
```

Description: An index and counter for events to be processed into the scenario file.

Package: ATSG_Gen_Pkg spec.

Usage: ATSG Global Data.

```
type Event_File_Type (Event_Kind : ATSG_Gen_Pkg.Event_Kind_Type :=
                                              ATSG_Gen_Pkg.ESM) is
 record
   Event_Time           : ATSG_Gen_Pkg.Event_Time_Type := -1.0;
   case Event_Kind is
     when  ATSG_Gen_Pkg.Operator_Input =>
       OIT_Action         : Operator_Pkg.Action_Type;
       OIT_Comment        : Operator_Pkg.Comment_Type;
     when  ATSG_Gen_Pkg.Situation_Response    =>
       SRT_Action         : Operator_Pkg.Situation_Response_Type;
     when others   =>
       Target_ID          :  ATSG_Gen_Pkg.Target_Num_Type;
       Q_Class_Err_Flg    : boolean;  -- Class error flag
       R_OOR_Flag         : boolean;  -- Out_of_range flag
   end case;
 end record;
```

Description: A record of the event to be processed into the scenario file.

Package: Event_Table_Obj_Pkg spec.

Usage: ATSG Global Data Type.

```
type Event_Kind_Type is
(Manual_Radar, TWS_Radar, ESM,
MAD -- _MARK, MAD_ACCEPT, MAD_ACCEPT1, MAD_ACCEPT2 (together)
, ACOUSTICS, IRDS,
VISUAL, COMM, OPERATOR_INPUT, SITUATION_RESPONSE);
```

Description: The category of the event.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

```
type Event_Table_Type is array (Event_Index_Type) of Event_File_Type;
```

Description: Table of events used by senario_file event.
Package: Event_Table_Obj_Pkg body.
Usage: Package Global Data Type.

```
type GAC_TYPE is
  record
    LAT              : LAT_TYPE;
    SIN_OF_LAT       : FLOAT;
    COS_OF_LAT       : FLOAT;
    LON              : LON_TYPE;
    SIN_OF_LON       : FLOAT;
    COS_OF_LON       : FLOAT;
  end record;
```

Description: Used to define information about the Gaming Area Center.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

```
type GAC_XY_TYPE is
  record
    X_POS   : TARGET_POS_ARRAY_TYPE ; -- ft
    Y_POS   : TARGET_POS_ARRAY_TYPE ; -- ft
    Z_POS   : TARGET_POS_ARRAY_TYPE ; -- ft
    XP_POS  : TARGET_POS_ARRAY_TYPE ; -- ft
    YP_POS  : TARGET_POS_ARRAY_TYPE ; -- ft
  end record;
```

Description: Used to define information about targets relative to the Gaming Area Center.

Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type INPUT_CATEGORY_TYPE is (LINE, ARC, INIT, THE_END);
```

Description: Used to define information about the Input Record.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

```
type INPUT_RECORD_TYPE(INPUT_CATEGORY: INPUT_CATEGORY_TYPE := LINE)
is
  record
    TARGET_NUM        : ATSG_Gen_Pkg.Target_Num_Type;
    SEGMENT_ID        : SEGMENT_ID_TYPE;
    case INPUT_CATEGORY is
      when INIT =>
        LAT               : FLOAT; -- Degrees
        LON               : FLOAT; -- Degrees
        ALT_DEP           : FLOAT; -- ft (neg is altitude)
        SPEED             : FLOAT; --Knots
     when LINE =>
        DURATION          : ATSG_Gen_Pkg.Event_Time_Type;  -- seconds
        HEADING           : FLOAT; -- Degrees
        AC_DC_RATE        : FLOAT;  -- Ft/sec
      when ARC =>
        RADIUS            : FLOAT; -- NM
        DIRECTION         : DIRECTION_TYPE;
        NO_OF_LOOPS       : INTEGER;
      when others =>
        MAN_RDR_SS        : FLOAT; -- ft
        TWS_RDR_SS        : FLOAT; -- ft
        ESM_SS            : FLOAT; -- ft
        MAD_SS            : FLOAT; -- ft
        ACOUSTIC_SS       : FLOAT; -- ft
        IRDS_SS           : FLOAT; -- ft
        VISUAL_SS         : FLOAT; -- ft
        COMM_SS           : FLOAT; -- ft
    end case;
  end record;
```

Description: Used to define the Input Record.

Package: PLATFORM_MOVEMENT_MODEL spec.

Usage: ATSG Global Data Type.

```
type Model_Kind_Type is (Deterministic, Probabilistic);
```

Description: The type of model being run.

Package: ATSG_Gen_Pkg spec.

Usage: ATSG Global Data Type.

```
type Object_Count_Type is array (ATSG_Gen_Pkg.Target_Index_Type)
                                        of ATSG_Gen_Pkg.Object_ID_Type;
```

Description: The object ID's to be assigned for each contact for a particular sensor. Start with 2 and increment. 0 is interpreted as an invalid update (contact not yet ADDED).

Package: Scenario_File_Obj_Pkg body.

Usage: Package Global Data Type.

```
type Platform_Array_Type is array (1 .. 3) of Platform_Type;
```

Description: An array of 3 possible NATO names for a target.

Package: ESM_Table_Obj_Pkg spec.

Usage: ATSG Global Data Type.

```
type PLATFORM_UPDATE_TYPE is array (Aircraft .. ATSG_Gen_Pkg.Max_Target) of
      FLOAT;
```

Description: Used to define the Platform Update.

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data Type.

```
type POSITION_INFO_TYPE is
  record
    -- A/C data
    AC_UPDATE_FLAG                  : UPDATE_TYPE;
    TRUE_AC_LAT                     : FLOAT; -- Radians
    TRUE_AC_LON                     : FLOAT; -- Radians
    AC_HEADING                      : FLOAT; -- Radians
    AC_ALTITUDE                   : FLOAT; -- Feet
    -- Target Data
    TGT_UPDATE_FLAG               : UPDATE_TYPE;
    TRUE_TARGET_LAT               : FLOAT; -- Radians
    TRUE_TARGET_LON               : FLOAT; -- Radians
    PERTURBED_TARGET_LAT          : FLOAT; -- Radians
    PERTURBED_TARGET_LON          : FLOAT; -- Radians
    AC_TO_TARGET_FLAT_RNG         : FLOAT; -- ft
    AC_TO_TARGET_SLANT_RNG        : FLOAT; -- ft
    AC_TO_TARGET_BEARING          : FLOAT; -- Radians
    end record;
```

Description: Used to define the Platform Positional Data.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

```
type Probability_Type is digits 9 range 0.0 .. 1.0;
```

Description: General purpose probability.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

```
type Response_Object_ID_Array_Type is array
            (Operator_Pkg.Situation_Response_Type) of Number_Of_Objects_Type;
```

Description: An indication of how many additional objects are generated by the responses.
Package: Object_Manager_Pkg spec.
Usage: ATSG Global Data Type.

```
type SEG_PTR is array ( INTEGER range <> ) of INTEGER;
```

Description: Used to define the Segment Pointer array.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type SI_OI_Indicator_Type is  (SI, OI, Neither);
```

Description: Indicates whether previous event selected was SI vs OI.
Package: Event_Table_Obj_Pkg.
Unit: Merge_Tables.
Usage: Local Data Type.

```
type Situation_Response_Type is
        (No_Action, Accept_Correl_Alert, Reject_Correl_Alert, Inhibit_Correl_Alert,
        Accept_Split_Track, Reject_Split_Track);
```

Description: A description of the SRT response actions available to the operator wrt sensor correlation/collocation.

Package: Operator_Pkg spec.

Usage: ATSG Global Data Type.

```
type STEERING_CMD_TYPE (COMMAND:COMMAND_TYPE := LINE) is
  record
    SPEED              : FLOAT; -- Ft/sec
    DURATION           : ATSG_Gen_Pkg.Event_Time_Type; -- seconds
    BEGIN_TIME         : ATSG_Gen_Pkg.Event_Time_Type; -- seconds
    case COMMAND is
      when LINE =>
        HEADING        : FLOAT; -- radians
        AC_DC_RATE     : FLOAT; -- Ft/sec
      when others => -- ARC
        END_HEADING    : FLOAT; -- radians
        RADIUS         : FLOAT; -- ft
        THETA          : FLOAT; -- radians
    end case;
  end record;
```

Description: Used to define the Steering Commands.

Package: PLATFORM_MOVEMENT_MODEL body.

Usage: Package Global Data Type.

```
type Target_Count_Array_Type is array (ATSG_Gen_Pkg.Target_Index_Type) of
                                                             Target_Count_Type;
```

Description: Count of contacts for all targets for all sensors. The first contact for each target is numbered 1.

Package: Free_Form_Obj_Pkg body.

Usage: Package Global Data Type.

```
type Target_End_Record_Type is
  record
    Used_Flag      : boolean := false;  -- indicates whether target is used
    Stringx        : string(1 .. 20);
    Stringy        : string(1 .. 20);
  end record;  -- Target_End_Record_Type
```

Description: Latest plot position (x, y) of a platform.
Package: Plot_File_Pkg body.
Usage: Package Global Data Type.

```
type TARGET_HDG_ARRAY_TYPE is array (Aircraft .. ATSG_Gen_Pkg.Max_Target) of
        FLOAT;
```

Description: Used to define the Target Heading.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type TARGET_POS_ARRAY_TYPE is array (Aircraft .. ATSG_Gen_Pkg.Max_Target) of
        FLOAT;
```

Description: Used to define the Target Position.
Package: PLATFORM_MOVEMENT_MODEL body.
Usage: Package Global Data Type.

```
type Target_Table_Type is array (ATSG_Gen_Pkg.Target_Index_Type) of
                                                    Emitter_Event_Type;
```

Description: The table of target types, classes, and number of emitters.
Package: ESM_Table_Obj_Pkg body.
Usage: Package Global Data Type.

```
type Update_Count_Type is array (ATSG_Gen_Pkg.Target_Index_Type) of integer;
```

Description: Count of contacts for all targets for a particular type of sensor. The first contact for each sensor is numbered 1 for each target. Different objects of this type are declared for each sensor.
Package: Scenario_File_Obj_Pkg body.
Usage: Package Global Data Type.

type UPDATE_TYPE is (UPDATE,DONT_UPDATE);
Description: Used to define the Target update status.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

subtype Comment_Type is string (1 .. 40);
Description: A string of instructions or information supporting the Operator Action_Type or Situation_Response_Type.
Package: Operator_Pkg spec.
Usage: ATSG Global Data Type.

subtype Emitter_Number_Type is Emitter_Count_Type range 0 .. MAXEMITTERS;
Description: The index of the ESM receptions for a target. 0 is N/A
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data Type.

subtype Event_Index_Type is ATSG_Gen_Pkg.Event_Count_Type
range 1 .. ATSG_Gen_Pkg.MAXEVENTS;
Description: Index of table of events used by senario_file.
Package: Event_Table_Obj_Pkg body.
Usage: Package Global Data Type.

subtype Event_Index_Type is Event_Count_Type range 1 .. Max_DOIT_Events;
Description: Index of DOIT event being used.
Package: D_OIT_Obj_Pkg body.
Usage: Package Global Data Type.

subtype Event_Index_Type is Event_Count_Type range 1 .. Max_DSIT_Events;
Description: Index of DSIT event being used.
Package: D_SIT_Obj_Pkg body.
Usage: Package Global Data Type.

subtype Event_Index_Type is Event_Count_Type range 1 .. Max_DSRT_Events;
Description: Index of DSRT event being used.
Package: D_SRT_Obj_Pkg body.
Usage: Package Global Data Type.

subtype Event_Time_Type is FLOAT range -1.0 .. float'last;
Description: Game time is seconds.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype LAT_TYPE is FLOAT range Min_LAT .. Max_LAT;
Description: The initial lat/lon of the P3 U4 aircraft defines the center of our gaming area. (0 X coordinate, 0 Y coordinate) Lat from -PI to PI (radians). In the NED system North and East are Positive while South and West are Negative.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

subtype LON_TYPE is FLOAT range Min_LON .. Max_LON;
Description: The initial lat/lon of the P3 U4 aircraft defines the center of our gaming area (0 X coordinate, 0 Y coordinate). Lon from -2PI to 2PI (radians). In the NED system North and East are Positive while South and West are Negative.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

subtype Number_Of_Objects_Type is integer;
Description: The number of additional objects to generate.
Package: Object_Manager_Pkg spec.
Usage: ATSG Global Data Type.

subtype Number_Type is integer range 0 .. ATSG_Gen_Pkg.Max_Target;
Description: Platforms, where 0 is A/C.
Package: Plot_File_Pkg body.
Usage: Package Global Data Type.

subtype Object_ID_Type is integer;
Description: The object ID assigned to a data base object. A 0 is assigned for the ADD contact and the value is incremented for each update of the contact be the sensor.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype Percent_Type is FLOAT range 0.0 .. 1.0; -- as ratio (0.0 .. 1.0)
Description: A generic ratio type.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype Plat_List_Set_T_type is DBDTZE_Contact_Classification_Type_Include.
DBDTZE_PLATFORM_LIST_SET_TYPE_TYPE;
Description: Describes the contents of the platform name list. Both the UNIVERSAL and EMPTY lists are arrays of null strings but they are handled differently in classification scoring and fusion.
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data Type.

subtype Platform_Type is
DBDTZE_Contact_Classification_Type_Include.DBDTZE_NATO_Platform_Name_Type;
Description: A NATO class of a target (10 char string).
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data Type.

subtype Prob_P_Type is Probability_Type;
Description: The probability of the sensor detecting the target.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype Prob_Q_Type is Probability_Type;
Description: Probability for generating conflicting classification
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype Prob_R_Type is Probability_Type;
Description: Probability for out_of_range or false alarm.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype SEGMENT_ID_TYPE is INTEGER range 0 .. Max_Seg_ID;
Description: Defines the index limits of the movement model.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

subtype String_7_Type is string (1 .. MAXp1);
Description: The output string of Quote_Terminate.
Package: Pack_Strings_Pkg spec.
Usage: ATSG Global Data Type.

subtype String_Six_Type is string (1 .. MAXsize);
Description: The input string used for Quote_Terminate.
Package: Pack_Strings_Pkg spec.
Usage: ATSG Global Data Type.

subtype Target_Category_Type is DBDTZE_Contact_Classification_Type_Include.
DBDTZE_Platform_Type_Type;
Description: A description of the category of target (Subsurface, Surface, Unknown, Air).
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data Type.

subtype Target_Count_Type is integer;
Description: Count of contacts for a target for any sensors. The first contact for each target is numbered 1.
Package: Free_Form_Obj_Pkg body.
Usage: Package Global Data Type.

subtype Target_Index_Type is Target_Num_Type range 1 .. Max_Target;
Description: The index of the target.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype TARGET_NUM_TYPE is INTEGER range 0 .. Max_target;
Description: Target number. 0 will be the required P3 UIV aircraft, while 1 thru 6 are the air, surface and sub test targets.
Package: PLATFORM_MOVEMENT_MODEL spec.
Usage: ATSG Global Data Type.

subtype Target_Num_Type is integer range 0 .. Max_Target;
Description: Target number. Zero will be the required P3 UIV A/C while 1 through Max_Target are the air, surface, and sub targets.
Package: ATSG_Gen_Pkg spec.
Usage: ATSG Global Data Type.

subtype Threat_Class_Type is
    DBDTZE_Contact_Classification_Type_Include.DBDTZE_Platform_Class_Type;
Description: A description of the class of target (Friendly, Hostile, Unknown).
Package: ESM_Table_Obj_Pkg spec.
Usage: ATSG Global Data Type.